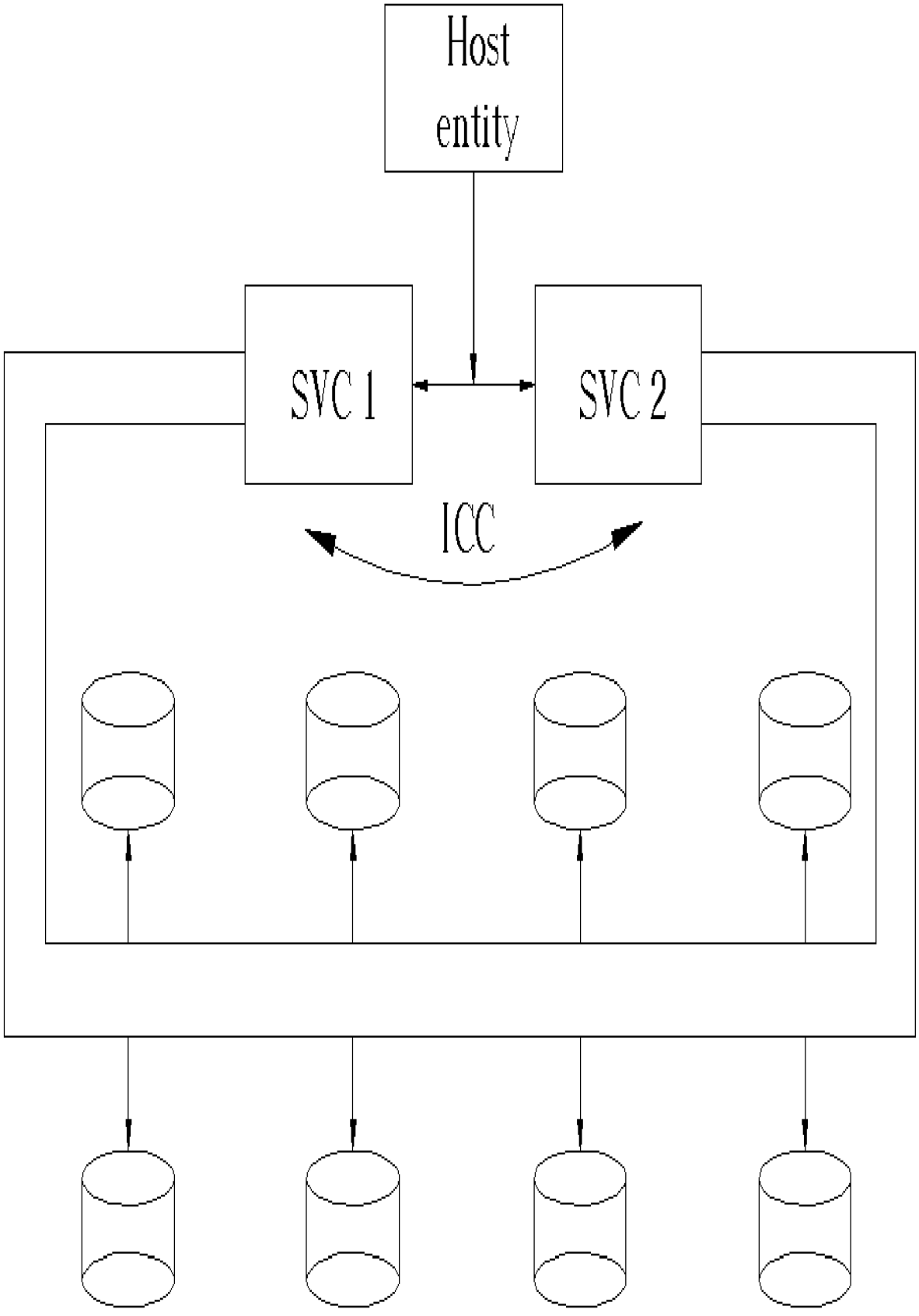


Fig. 1 Prior Art

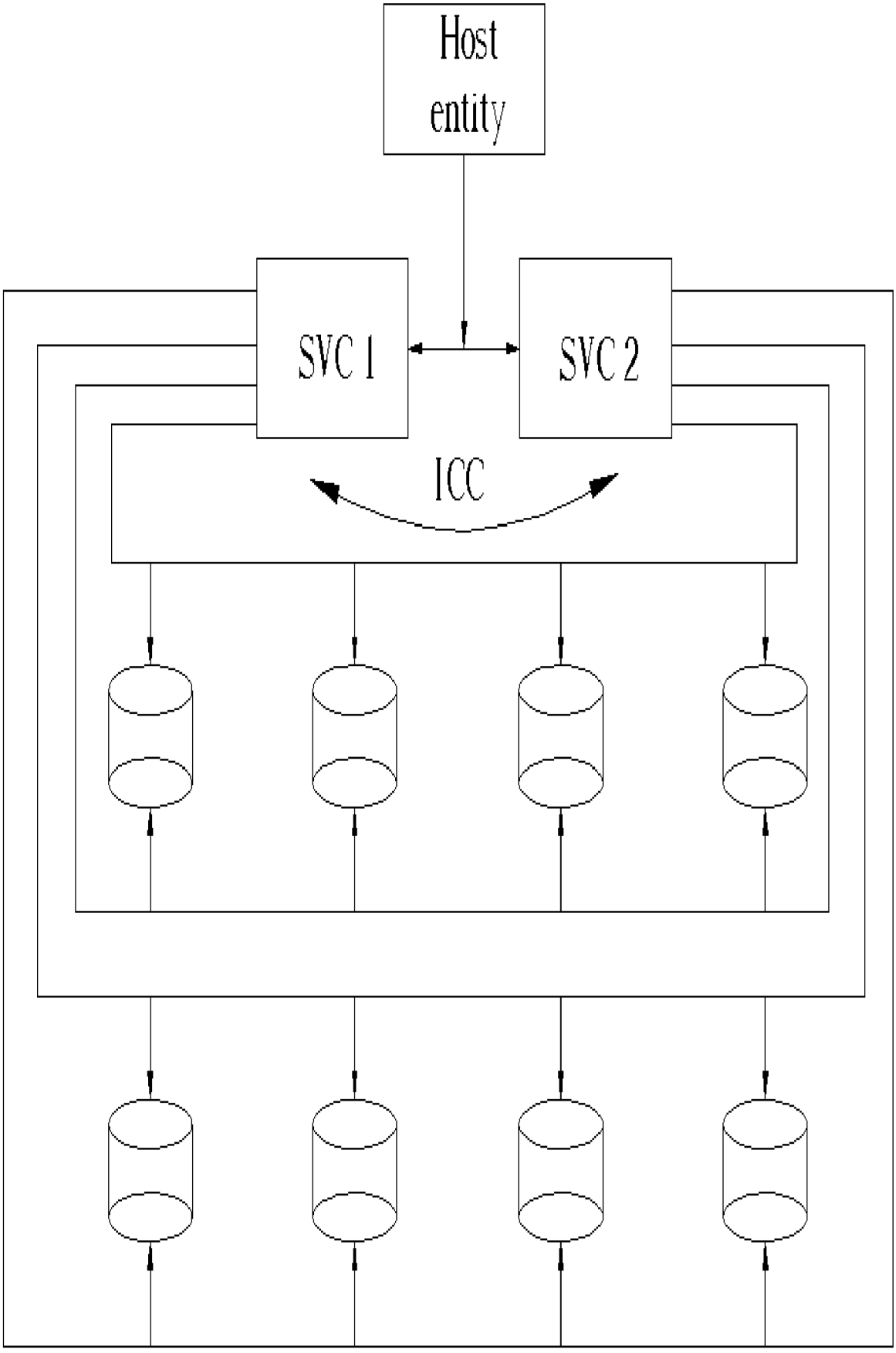


Fig. 2 Prior Art

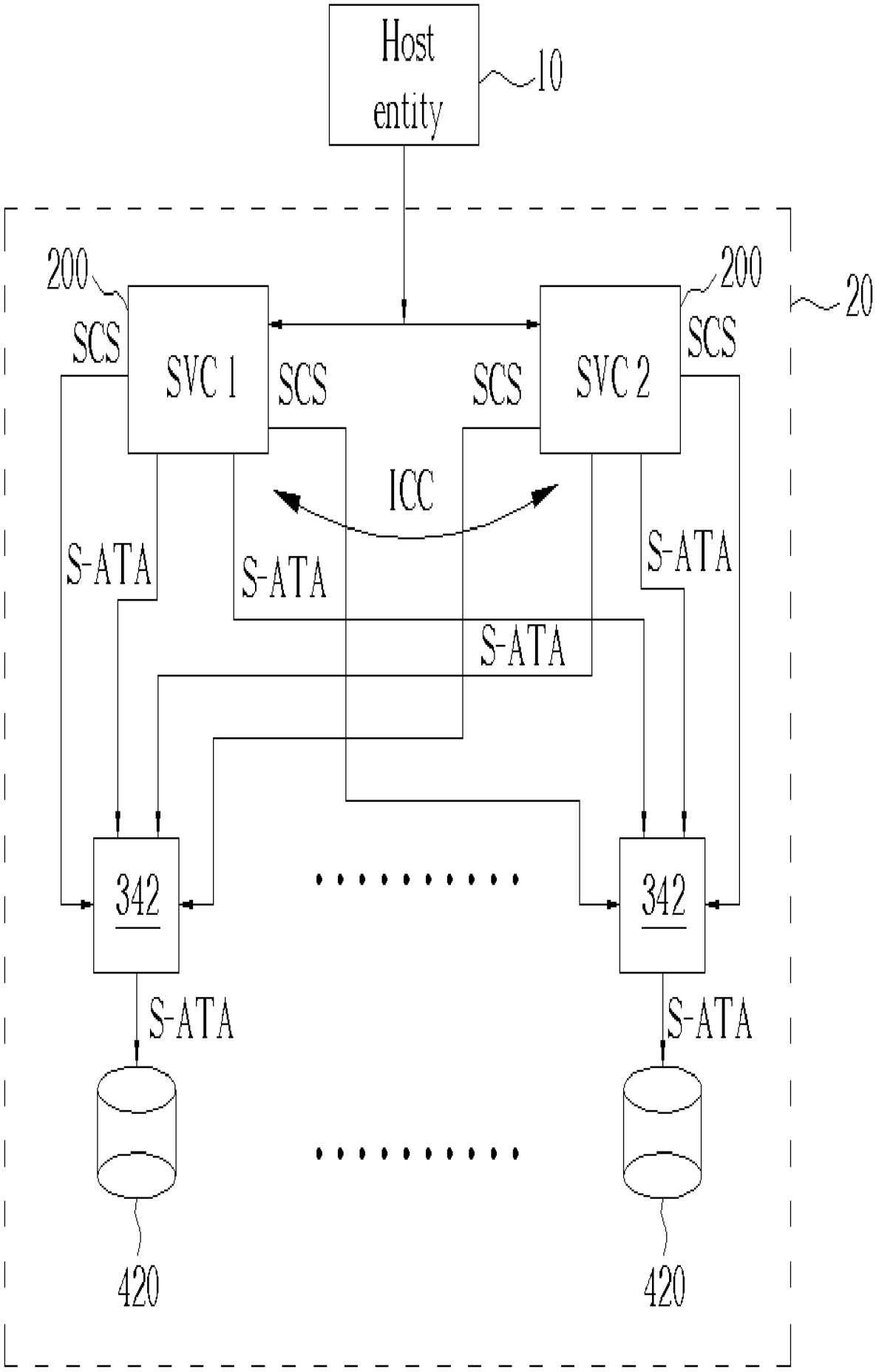

Host entity
10
200
200
20
SCS
SVC 1
SCS
SCS
SVC 2
SCS
ICC
S-ATA
S-ATA
S-ATA
S-ATA
342
342
S-ATA
S-ATA
420
420

Fig. 3

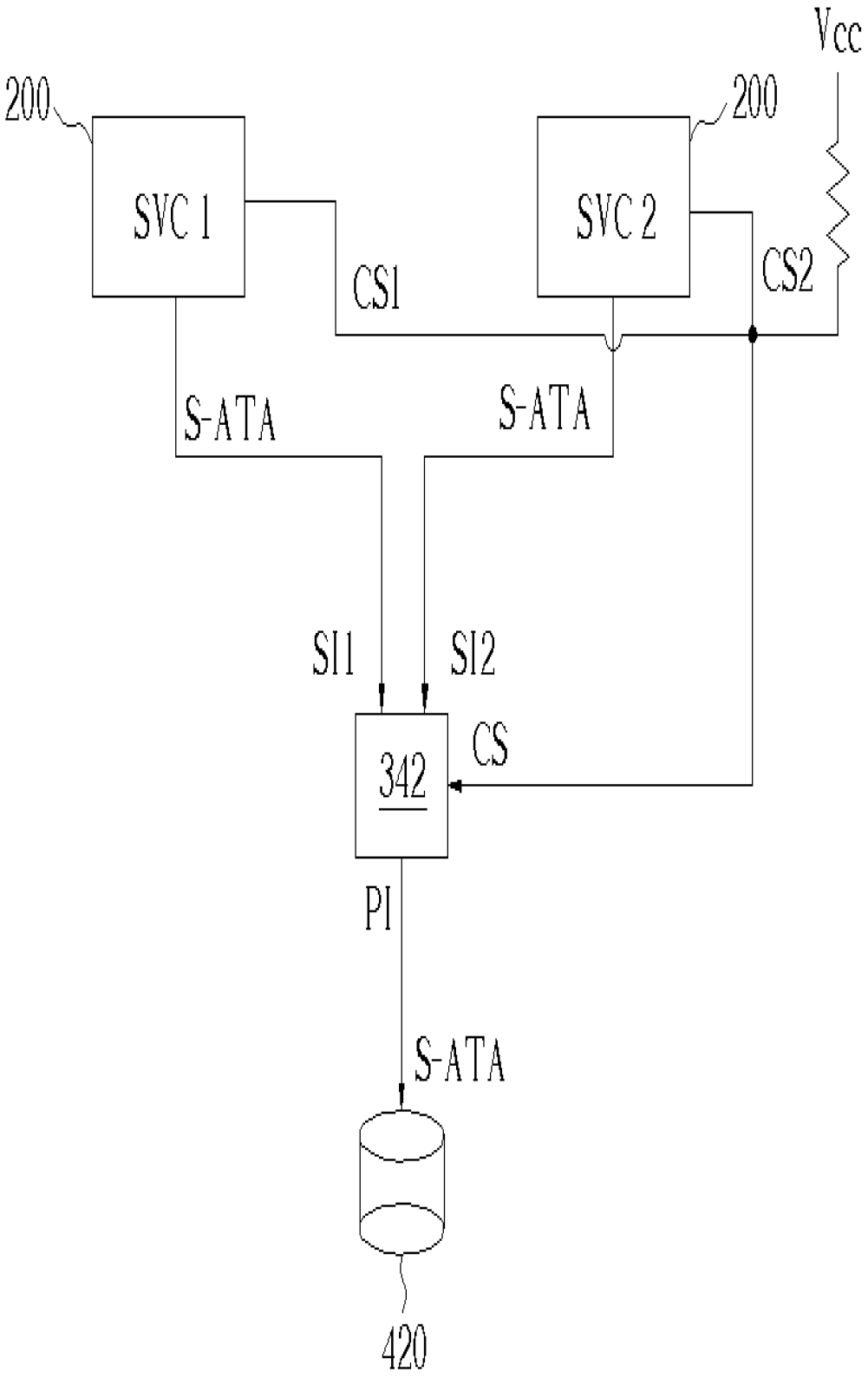

Vcc
200
200
SVC 1
SVC 2
CS1
CS2
S-ATA
S-ATA
SI1
SI2
342
CS
PI
S-ATA
420

Fig. 4

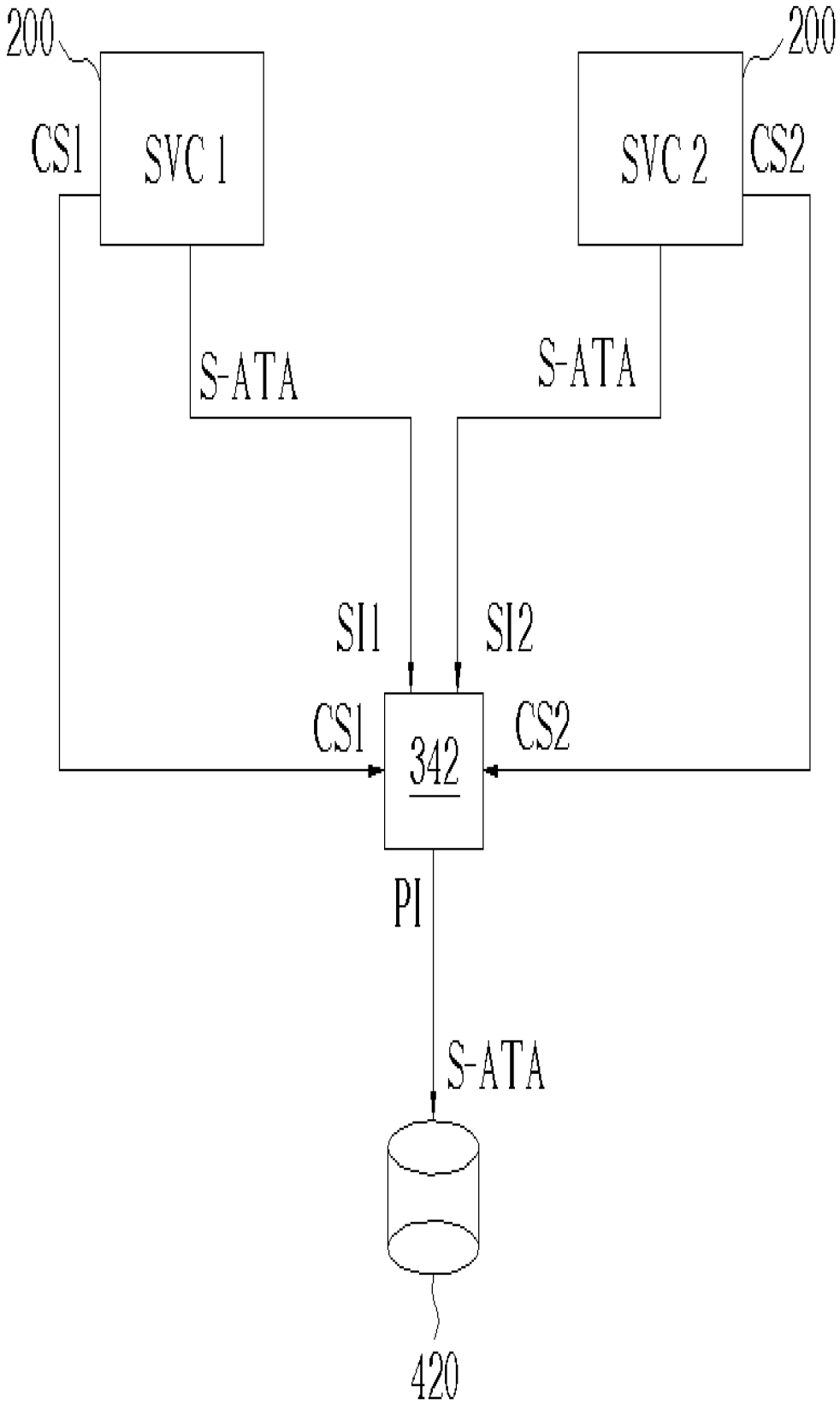

200
200
CS1
SVC 1
SVC 2
CS2
S-ATA
S-ATA
SI1
SI2
CS1
342
CS2
PI
S-ATA
420

Fig. 5

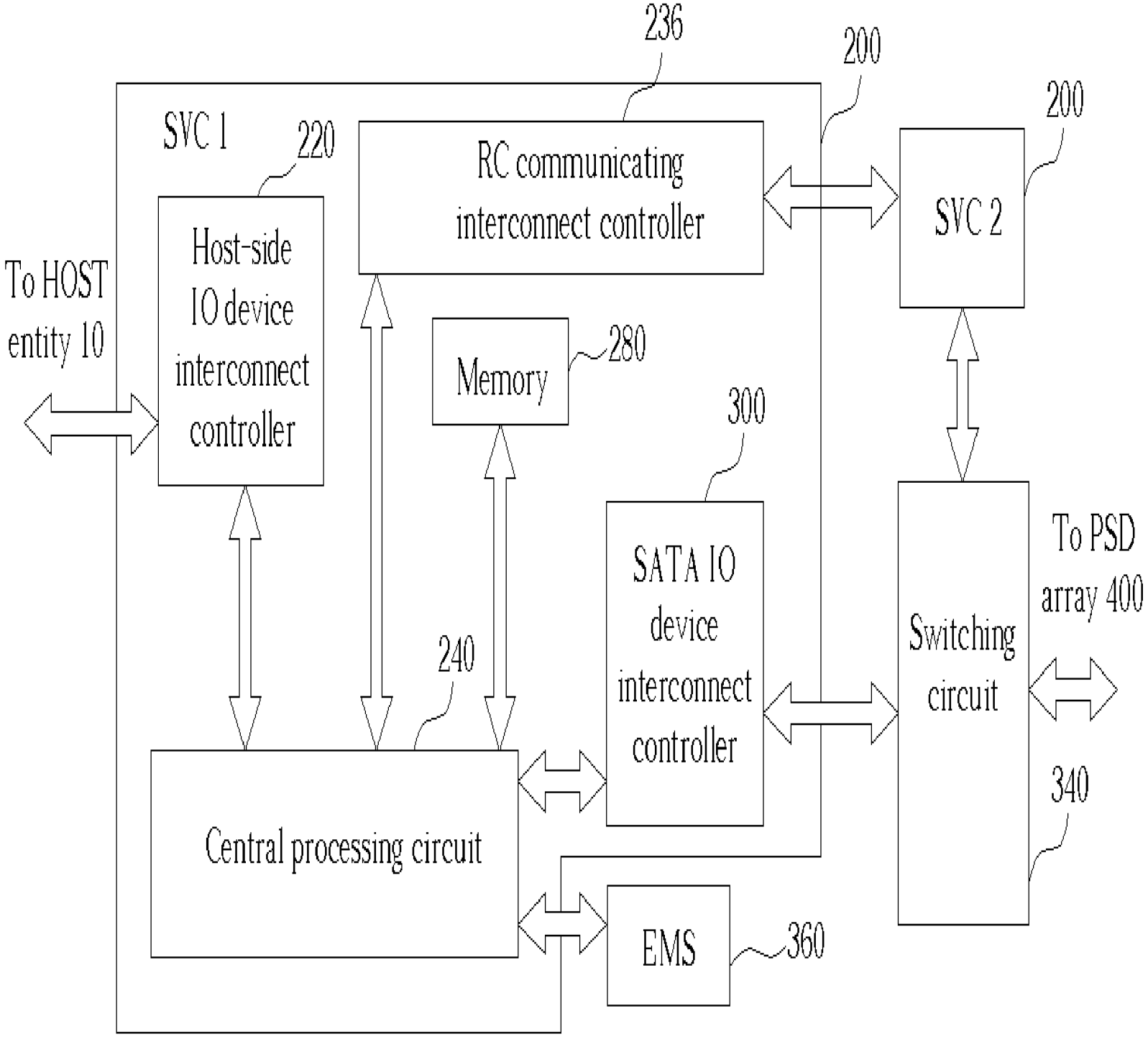

236
200
200
SVC 1
220
RC communicating interconnect controller
SVC 2
To HOST entity 10
Host-side IO device interconnect controller
Memory
280
300
SATA IO device interconnect controller
Switching circuit
To PSD array 400
240
340
Central processing circuit
EMS
360

Fig. 6

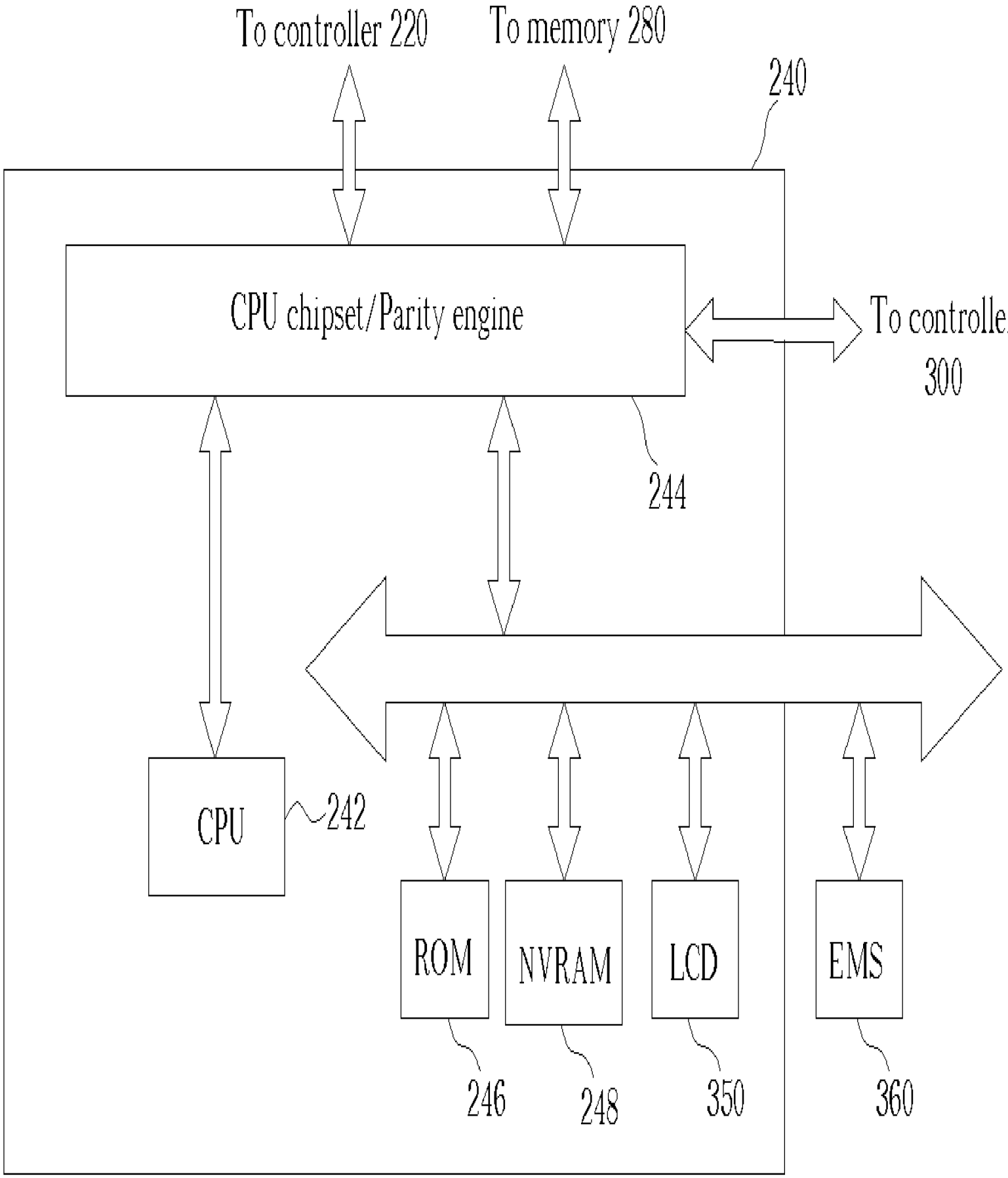

To controller 220
To memory 280
240
CPU chipset/Parity engine
To controller 300
244
CPU
242
ROM
NVRAM
LCD
EMS
246
248
350
360

Fig. 7

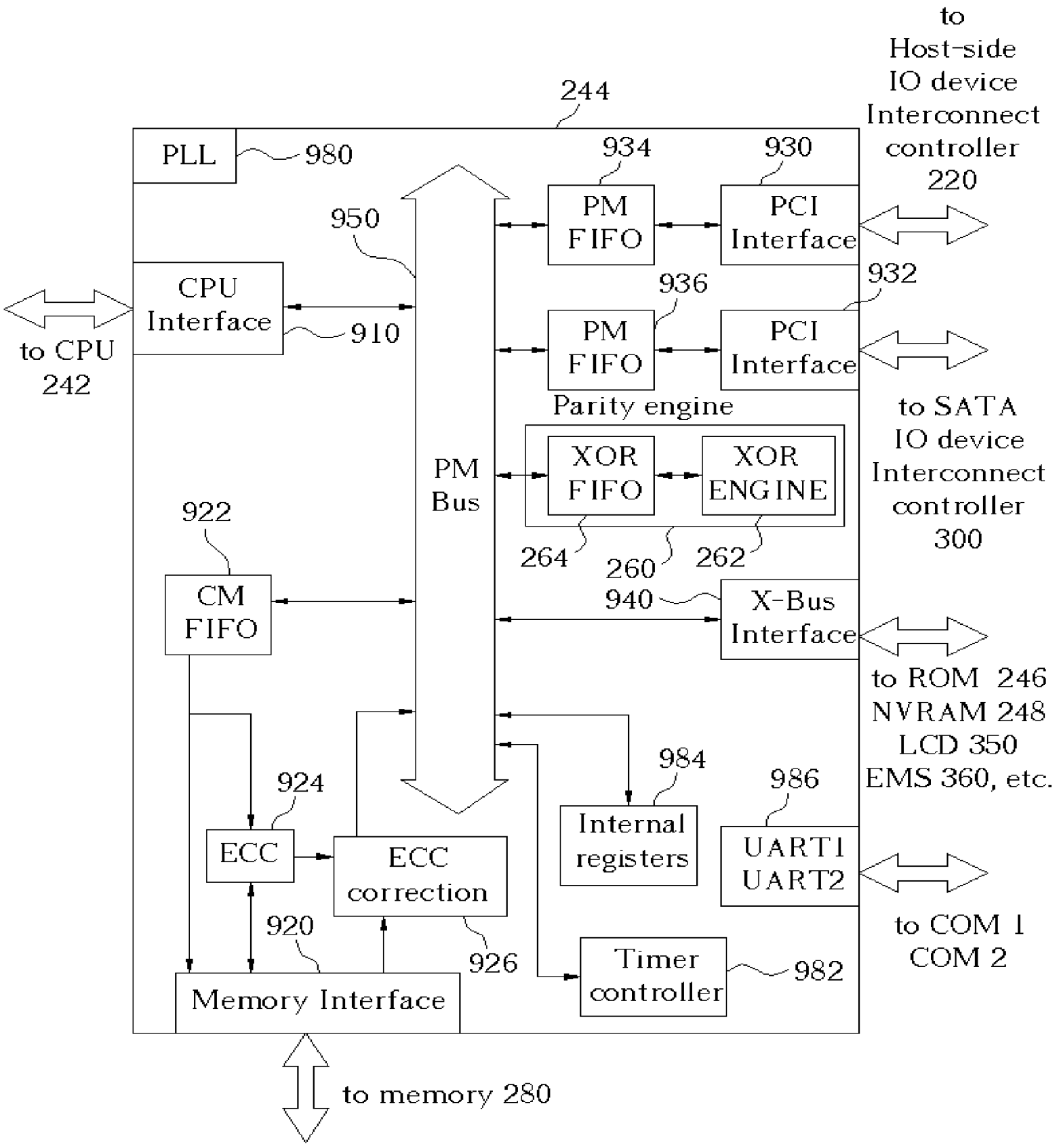

to
Host-side
IO device
Interconnect
controller
220
244
PLL
980
934
930
PM
FIFO
PCI
Interface
950
CPU
Interface
910
936
932
PM
FIFO
PCI
Interface
to CPU
242
Parity engine
to SATA
IO device
Interconnect
controller
300
PM
Bus
XOR
FIFO
XOR
ENGINE
922
264
260
262
CM
FIFO
940
X-Bus
Interface
to ROM 246
NVRAM 248
LCD 350
EMS 360, etc.
984
924
986
Internal
registers
ECC
ECC
correction
UART1
UART2
to COM 1
COM 2
920
926
Timer
controller
982
Memory Interface
to memory 280

Fig. 8

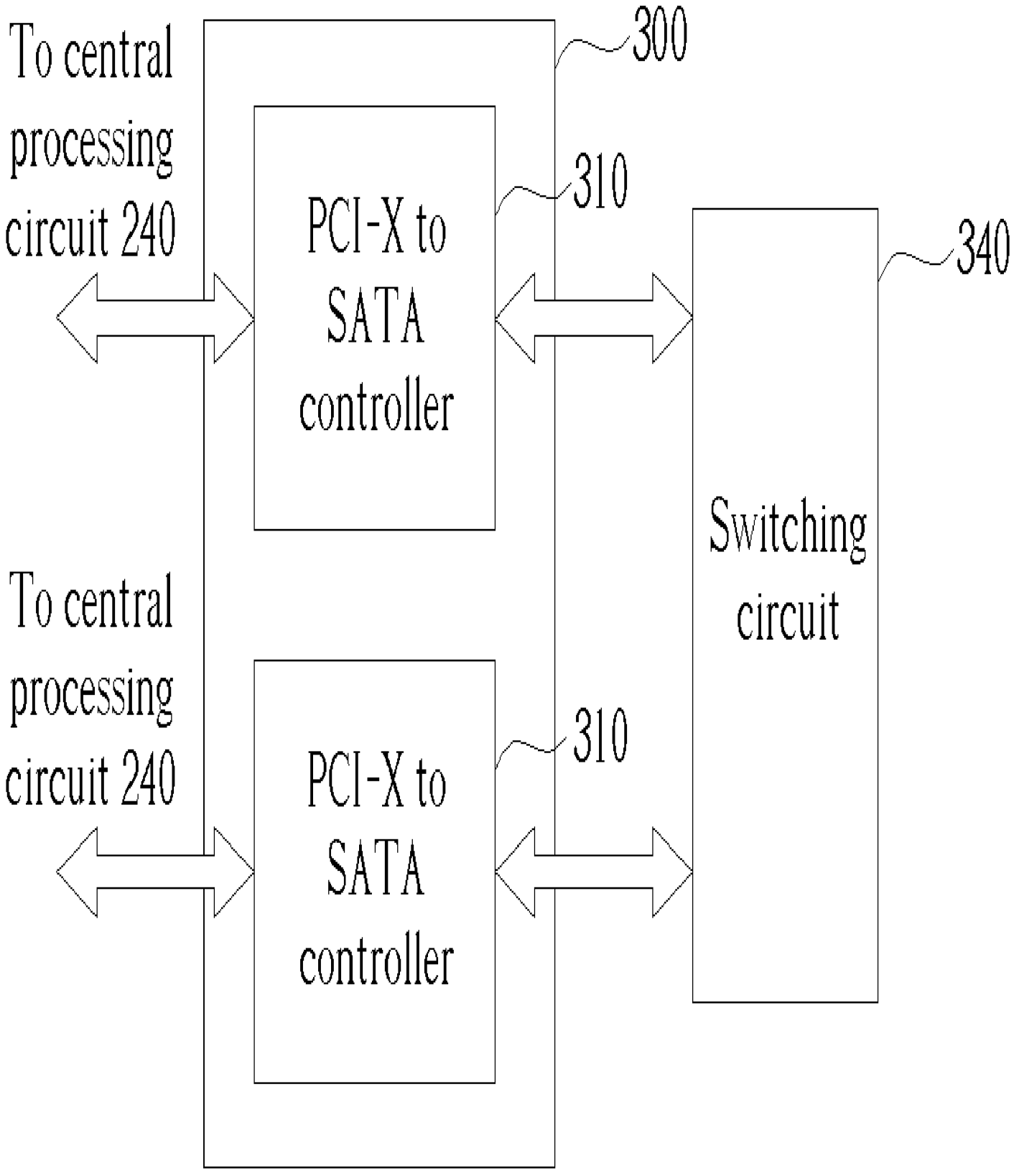

To central processing circuit 240
300
PCI-X to SATA controller
310
340
Switching circuit
To central processing circuit 240
PCI-X to SATA controller
310

Fig. 9

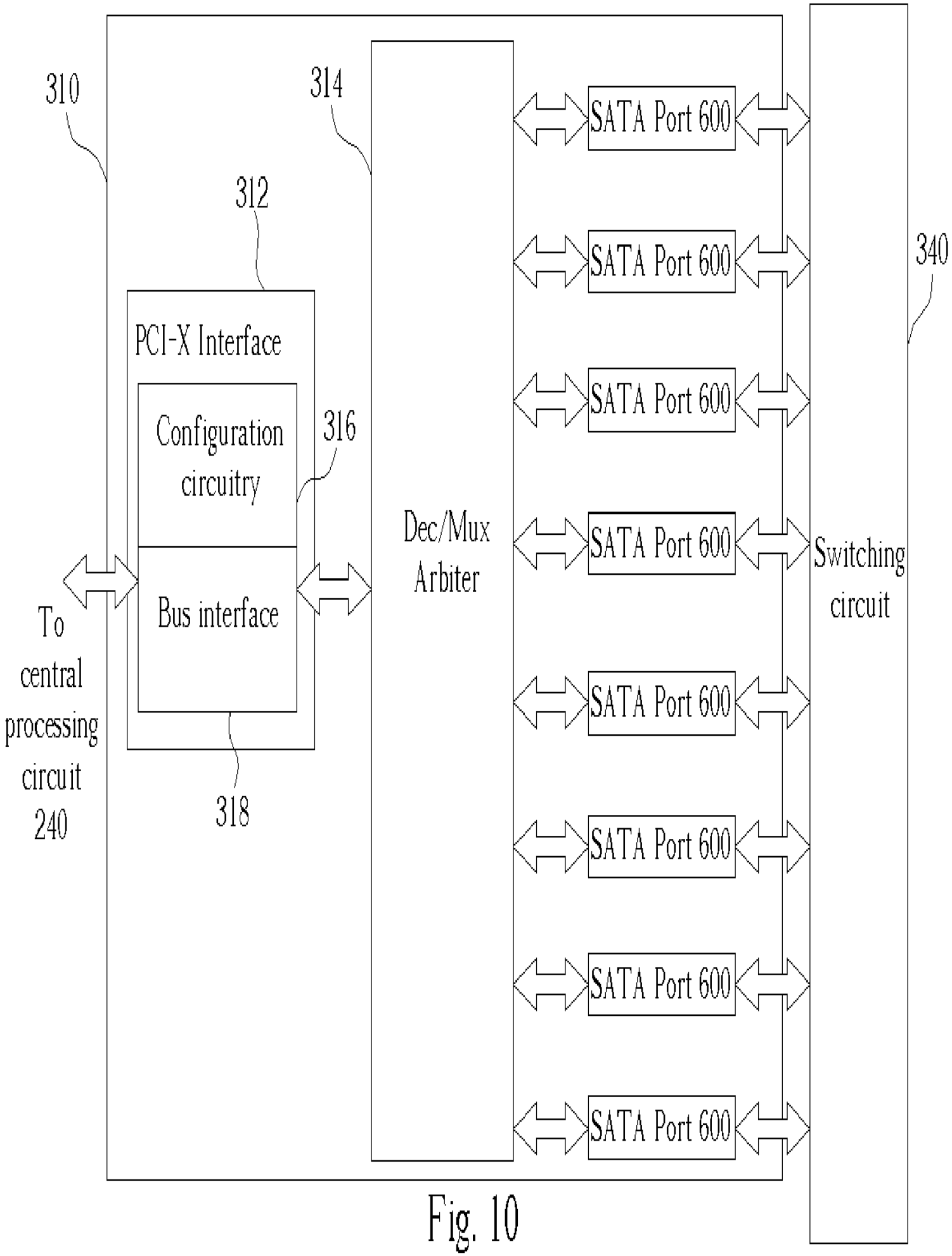

310
314
312
340
PCI-X Interface
Configuration circuitry
316
Bus interface
318
Dec/Mux Arbiter
SATA Port 600
SATA Port 600
SATA Port 600
SATA Port 600
SATA Port 600
SATA Port 600
SATA Port 600
SATA Port 600
Switching circuit
To central processing circuit 240

Fig. 10

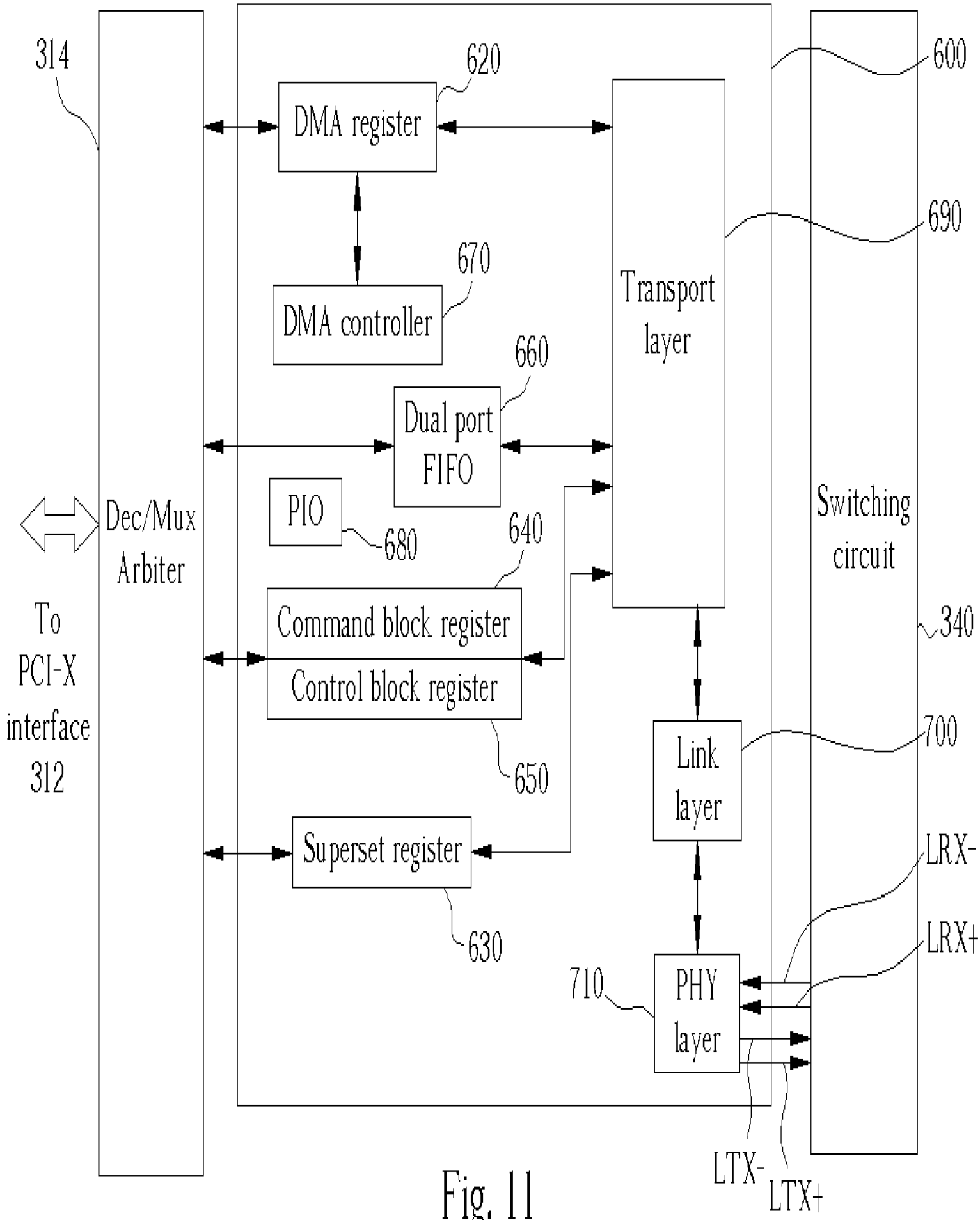

314
620
600
DMA register
690
670
DMA controller
Transport layer
660
Dual port FIFO
PIO
680
640
Dec/Mux Arbiter
Switching circuit
To PCI-X interface 312
Command block register
Control block register
340
650
700
Link layer
Superset register
LRX-
LRX+
630
710
PHY layer
LTX-
LTX+

Fig. 11

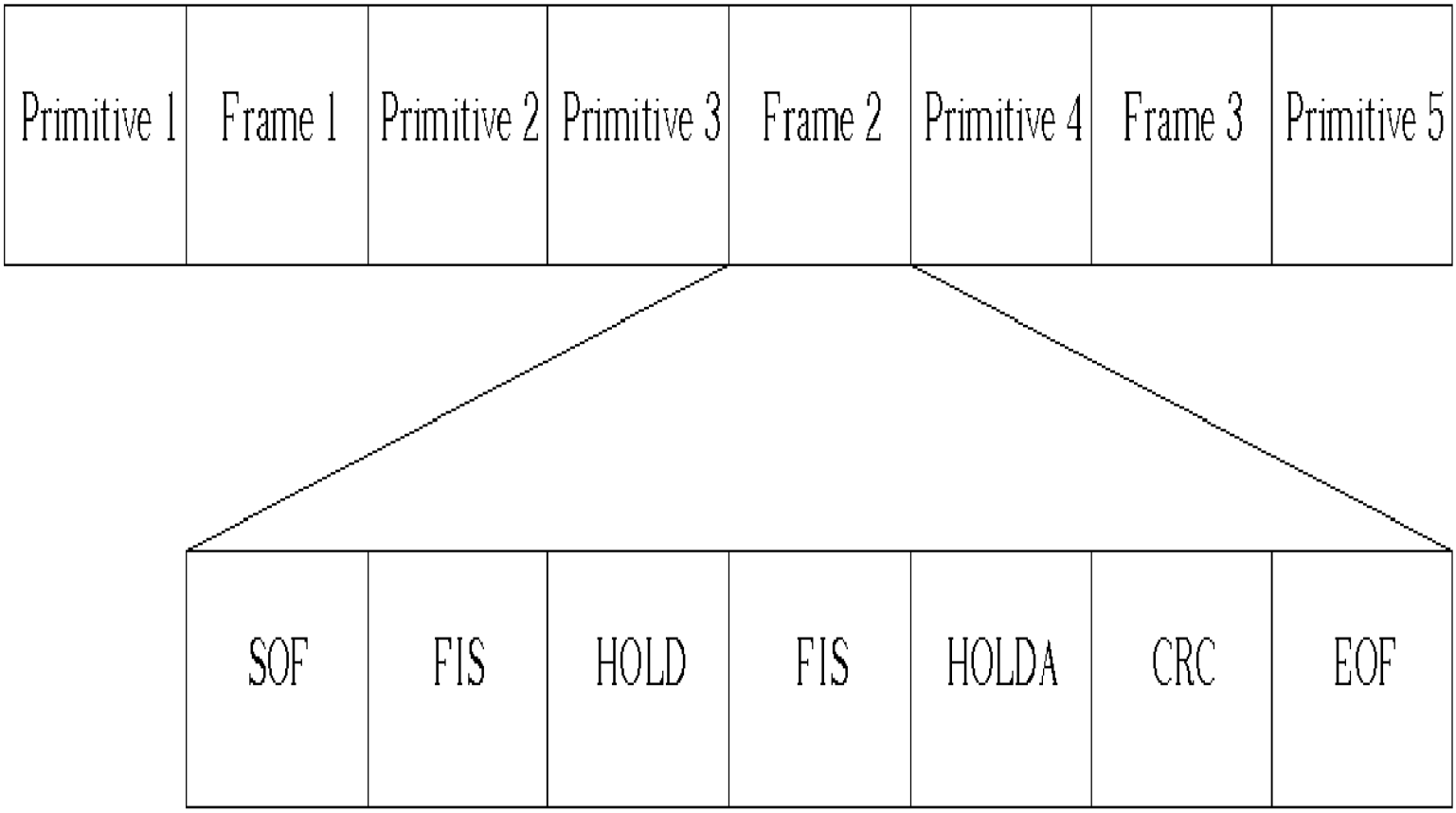

Primitive 1
Frame 1
Primitive 2
Primitive 3
Frame 2
Primitive 4
Frame 3
Primitive 5
SOF
FIS
HOLD
FIS
HOLDA
CRC
EOF

Fig. 12

0	Reserved (0)	Reserved (0)	R	I	D	Reserved (0)	FIS Type (41h)
1	DMA buffer identifier Low						
2	DMA buffer identifier High						
3	Reserved (0)						
4	DMA buffer offset						
5	DMA transfer count						
6	Reserved (0)						

Fig. 13

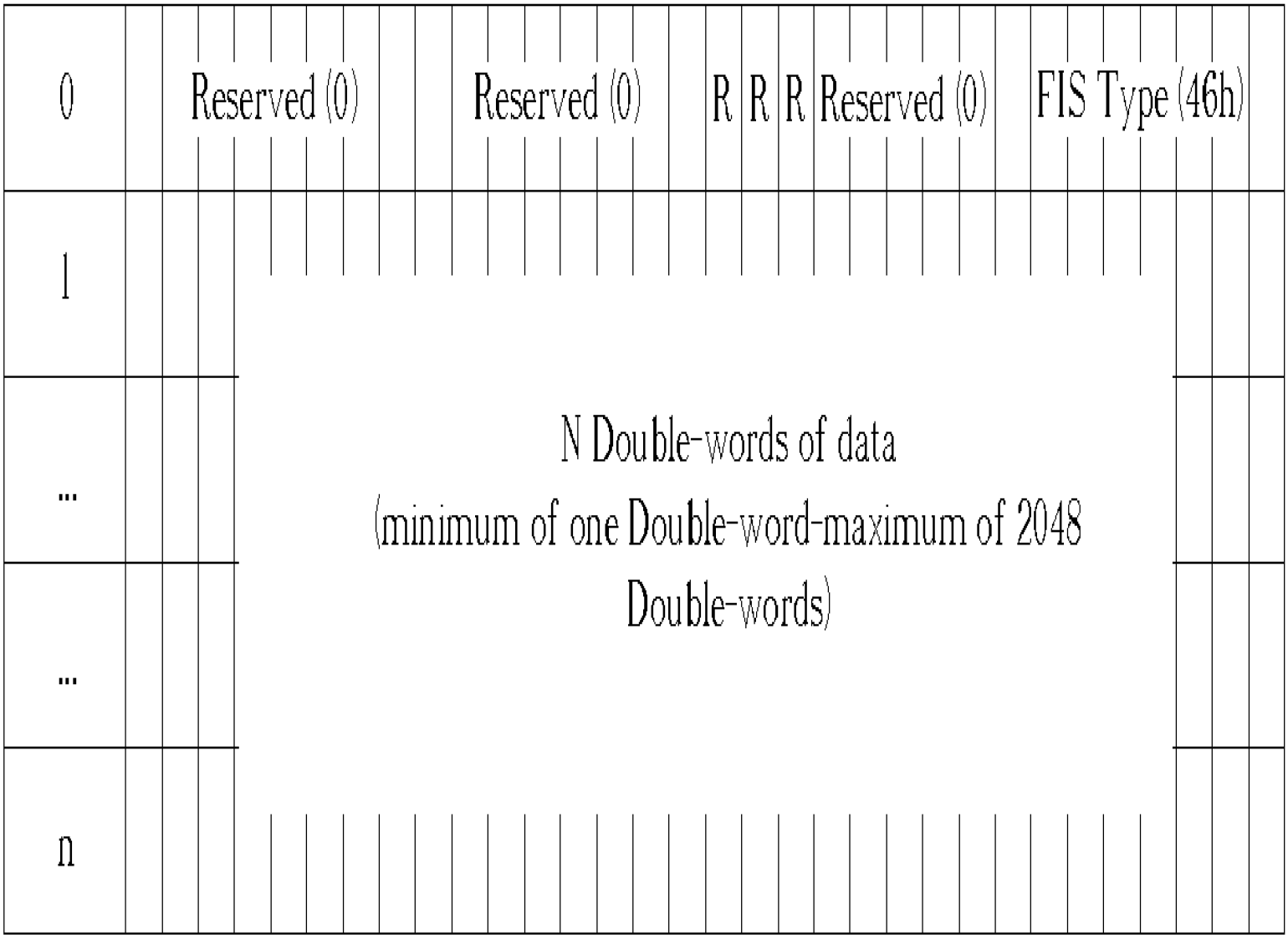

0
Reserved (0)
Reserved (0)
R R R Reserved (0)
FIS Type (46h)
1
N Double-words of data
(minimum of one Double-word-maximum of 2048
Double-words)
...
...
n

Fig. 14

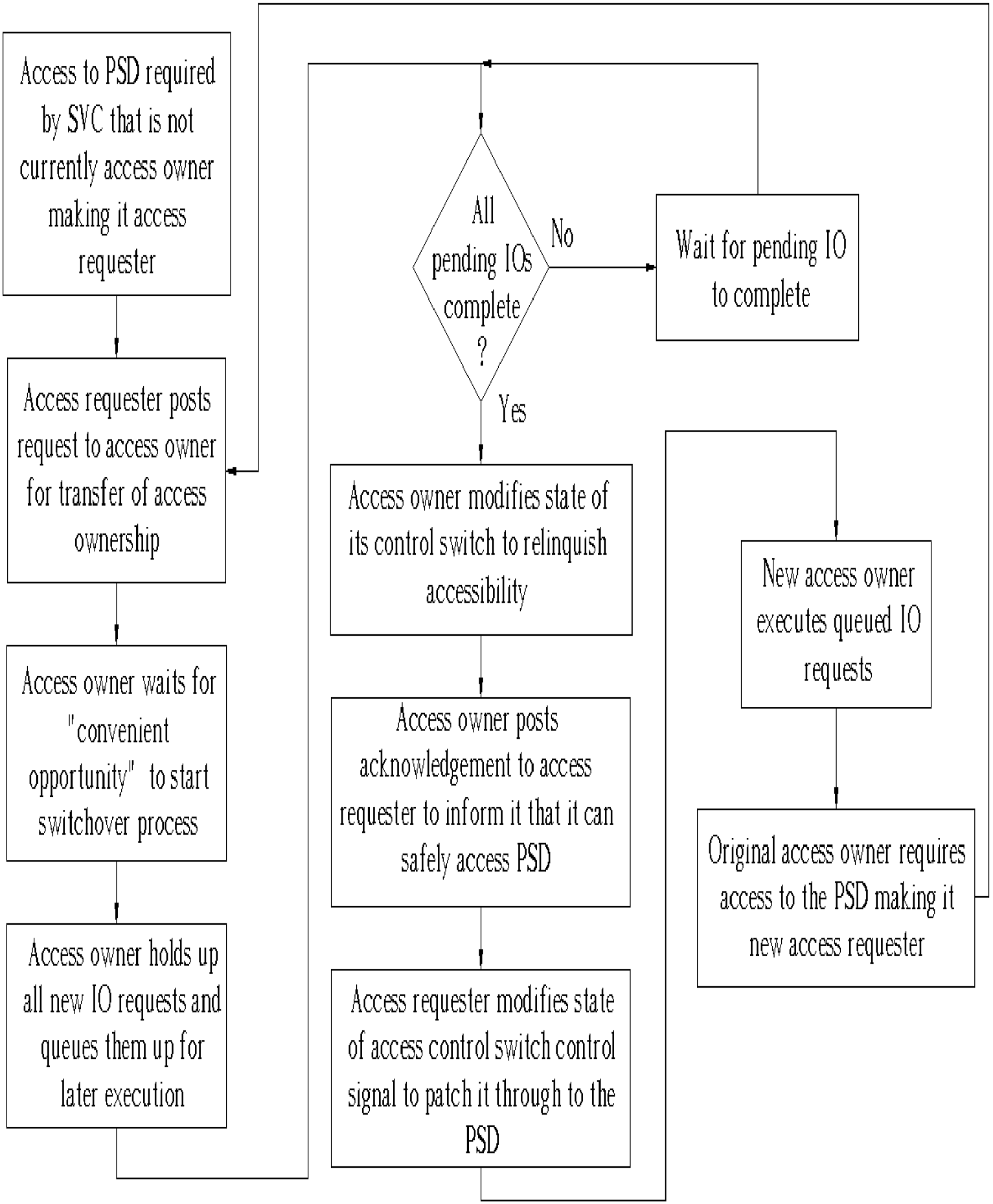

Access to PSD required by SVC that is not currently access owner making it access requester
Access requester posts request to access owner for transfer of access ownership
Access owner waits for "convenient opportunity" to start switchover process
Access owner holds up all new IO requests and queues them up for later execution
All pending IOs complete ?
No
Wait for pending IO to complete
Yes
Access owner modifies state of its control switch to relinquish accessibility
Access owner posts acknowledgement to access requester to inform it that it can safely access PSD
Access requester modifies state of access control switch control signal to patch it through to the PSD
New access owner executes queued IO requests
Original access owner requires access to the PSD making it new access requester

Fig. 15

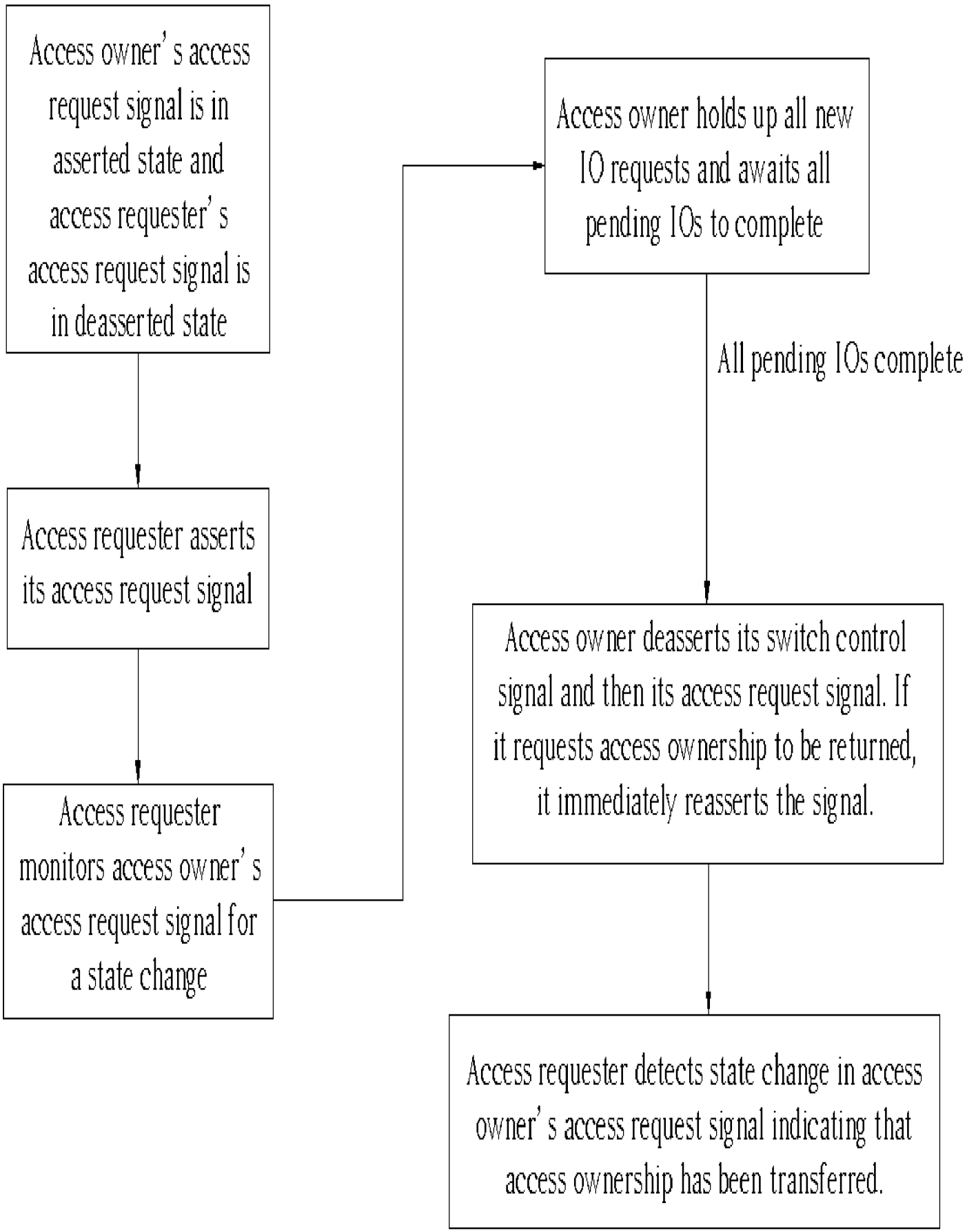
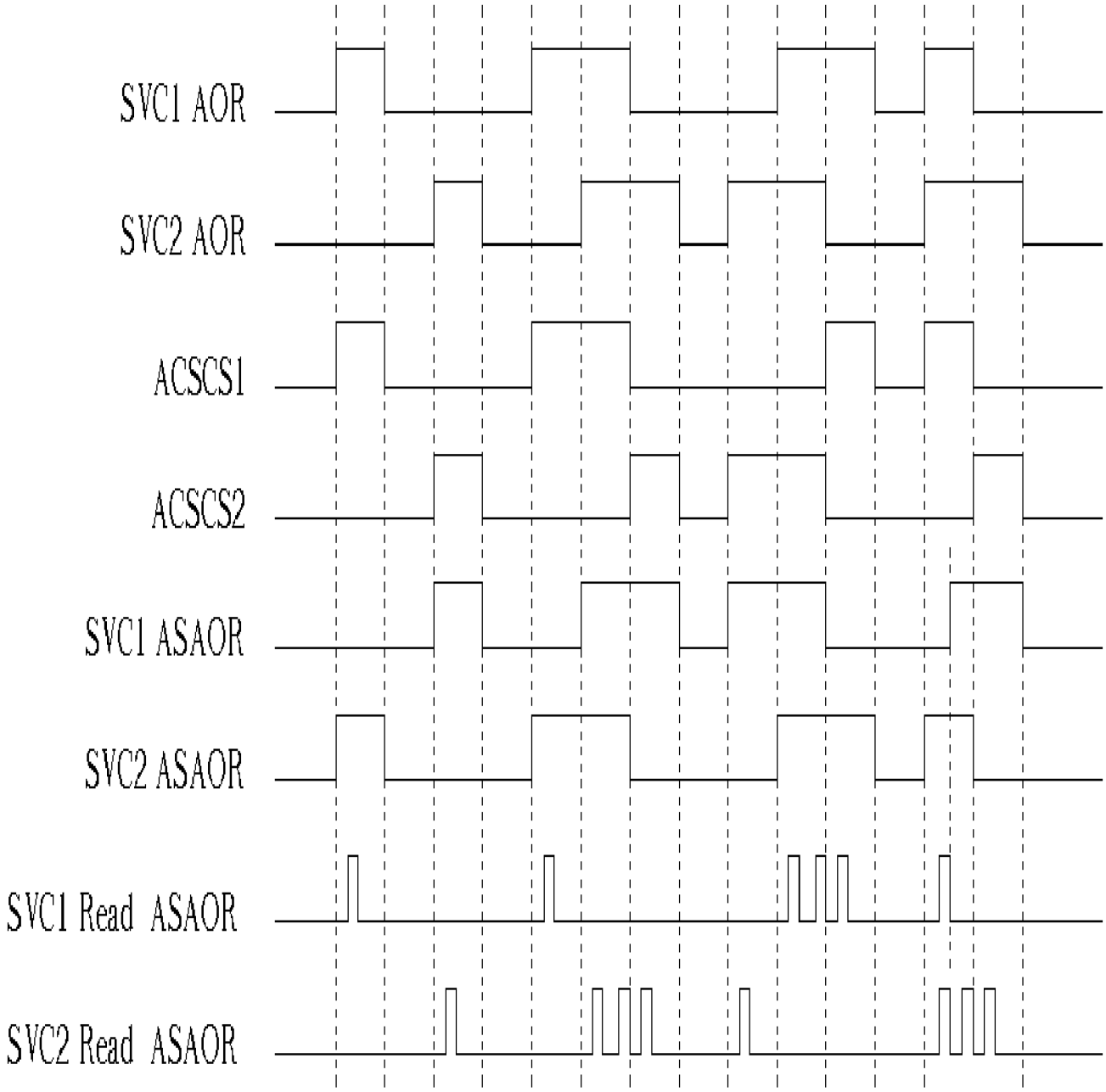

Access owner's access request signal is in asserted state and access requester's access request signal is in deasserted state
Access requester asserts its access request signal
Access requester monitors access owner's access request signal for a state change
Access owner holds up all new IO requests and awaits all pending IOs to complete
All pending IOs complete
Access owner deasserts its switch control signal and then its access request signal. If it requests access ownership to be returned, it immediately reasserts the signal.
Access requester detects state change in access owner's access request signal indicating that access ownership has been transferred.

Fig. 16



SVC1 AOR : SVC1's AOR Signal

ACSCS1 : SVC1's Access Control Switch Control Signal

SVC1 ASAOR : SVC1's Alternate SVC AOR Signal

SVC1 Read ASAOR : SVC1 Read ASAOR State Pulse

SVC2 AOR : SVC2's AOR Signal

ACSCS2 : SVC2's Access Control Switch Control Signal

SVC2 ASAOR : SVC2's Alternate SVC AOR Signal

SVC2 Read ASAOR : SVC2 Read ASAOR State Pulse

Fig. 17

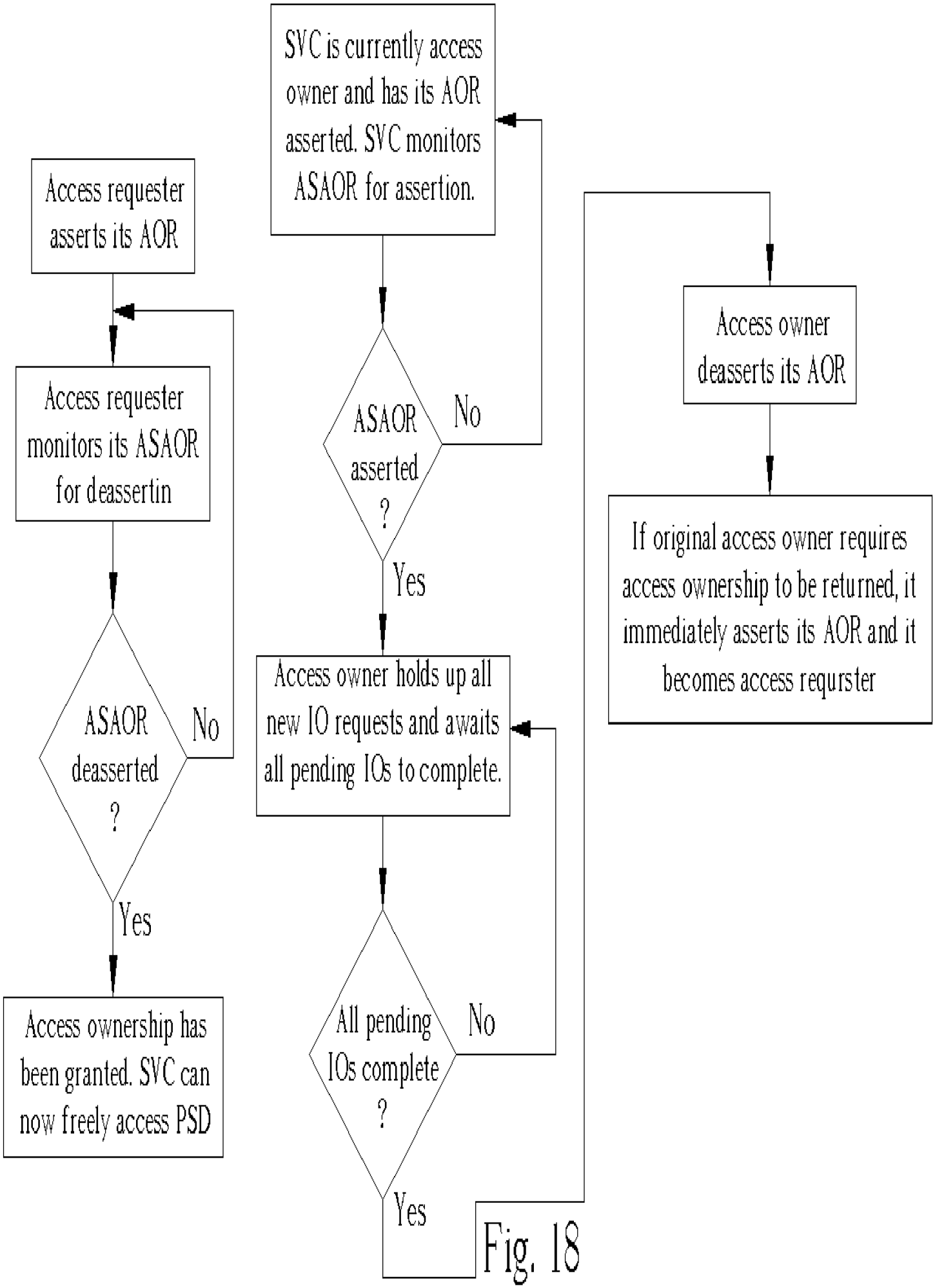

Access requester asserts its AOR
Access requester monitors its ASAOR for deassertin
ASAOR deasserted ?
No
Yes
Access ownership has been granted. SVC can now freely access PSD
SVC is currently access owner and has its AOR asserted. SVC monitors ASAOR for assertion.
ASAOR asserted ?
No
Yes
Access owner holds up all new IO requests and awaits all pending IOs to complete.
All pending IOs complete ?
No
Yes
Access owner deasserts its AOR
If original access owner requires access ownership to be returned, it immediately asserts its AOR and it becomes access requrster

Fig. 18

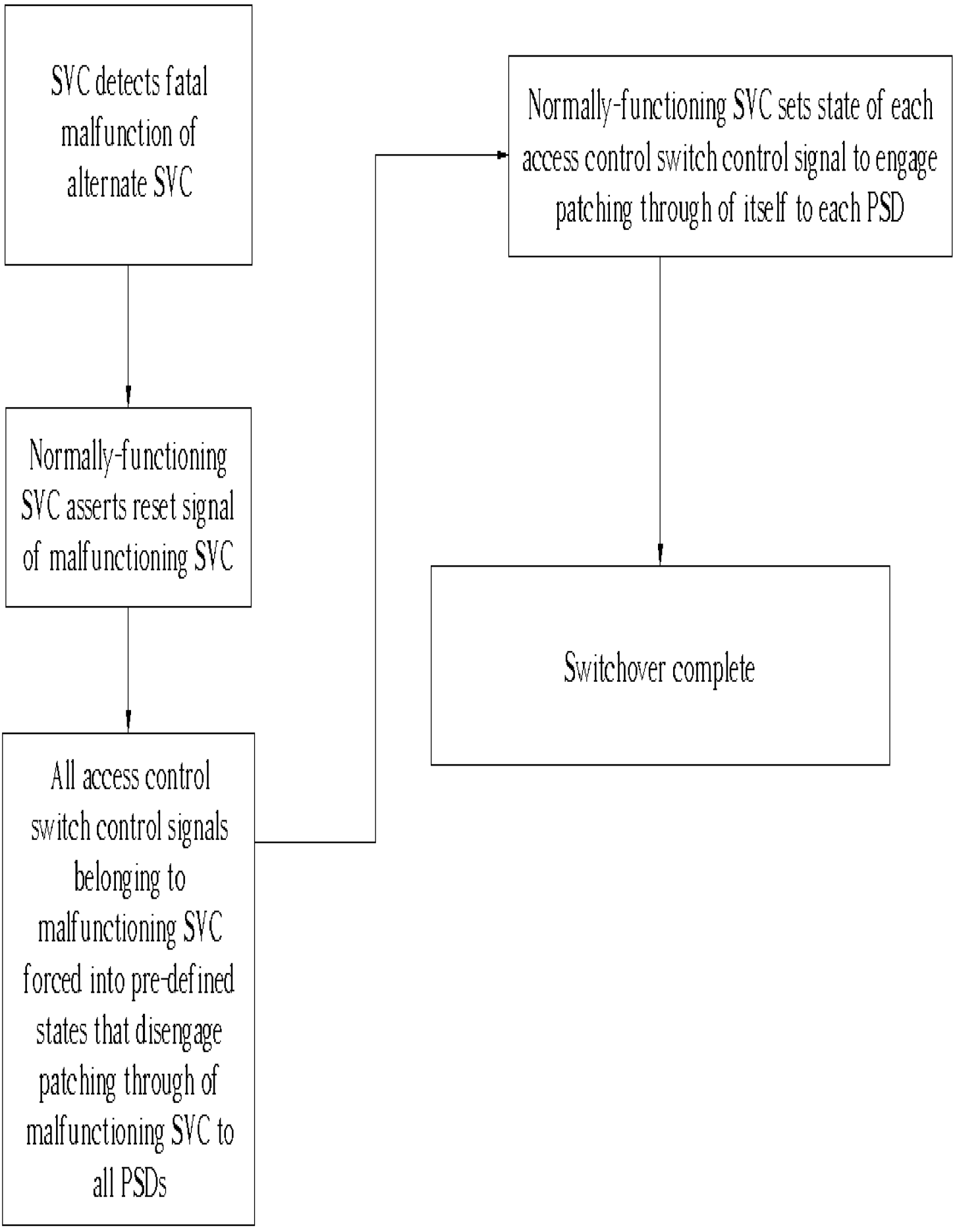

SVC detects fatal malfunction of alternate SVC
Normally-functioning SVC asserts reset signal of malfunctioning SVC
All access control switch control signals belonging to malfunctioning SVC forced into pre-defined states that disengage patching through of malfunctioning SVC to all PSDs
Normally-functioning SVC sets state of each access control switch control signal to engage patching through of itself to each PSD
Switchover complete

Fig. 19

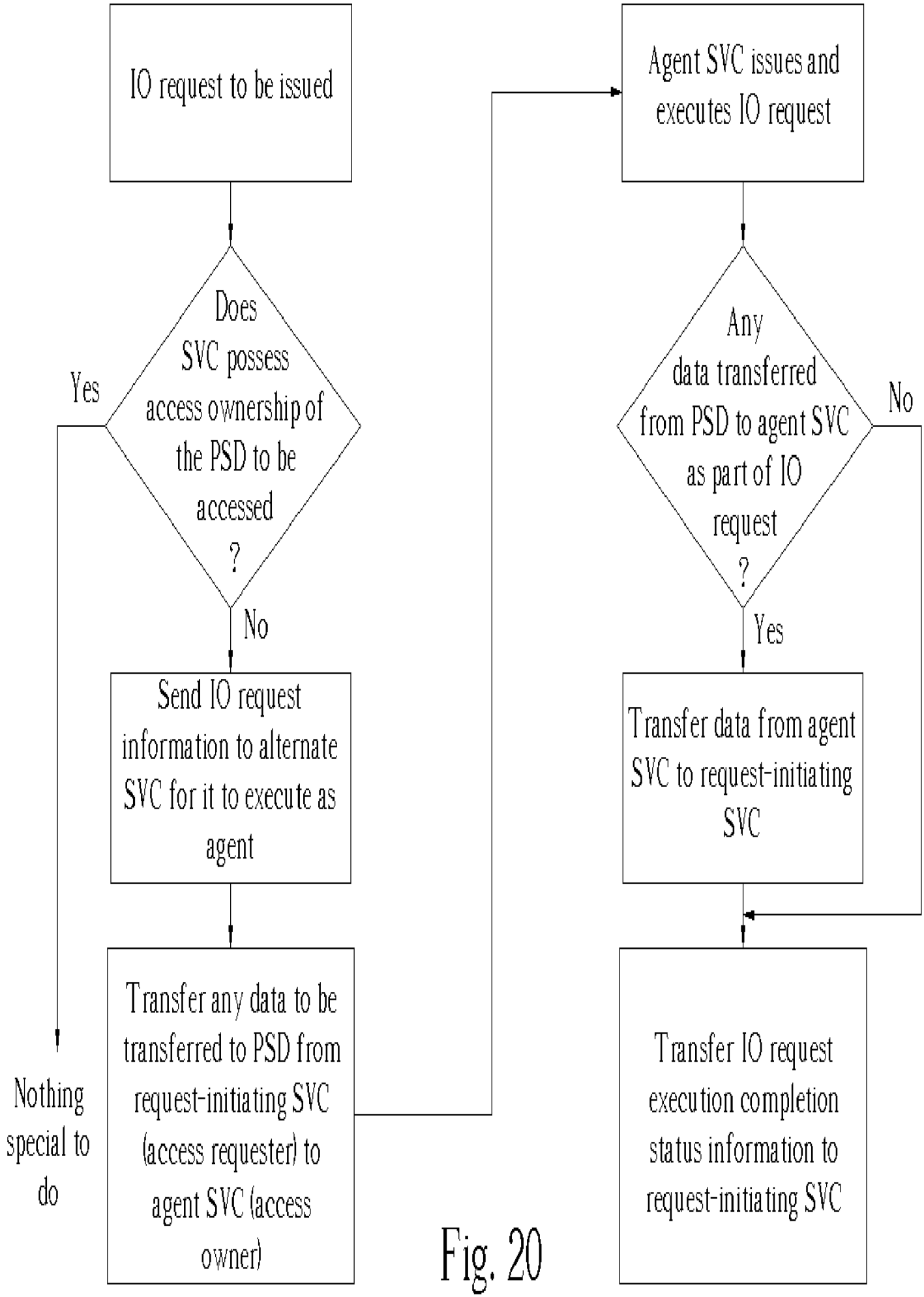

IO request to be issued
Does SVC possess access ownership of the PSD to be accessed ?
Yes
No
Nothing special to do
Send IO request information to alternate SVC for it to execute as agent
Transfer any data to be transferred to PSD from request-initiating SVC (access requester) to agent SVC (access owner)
Agent SVC issues and executes IO request
Any data transferred from PSD to agent SVC as part of IO request ?
Yes
No
Transfer data from agent SVC to request-initiating SVC
Transfer IO request execution completion status information to request-initiating SVC

Fig. 20

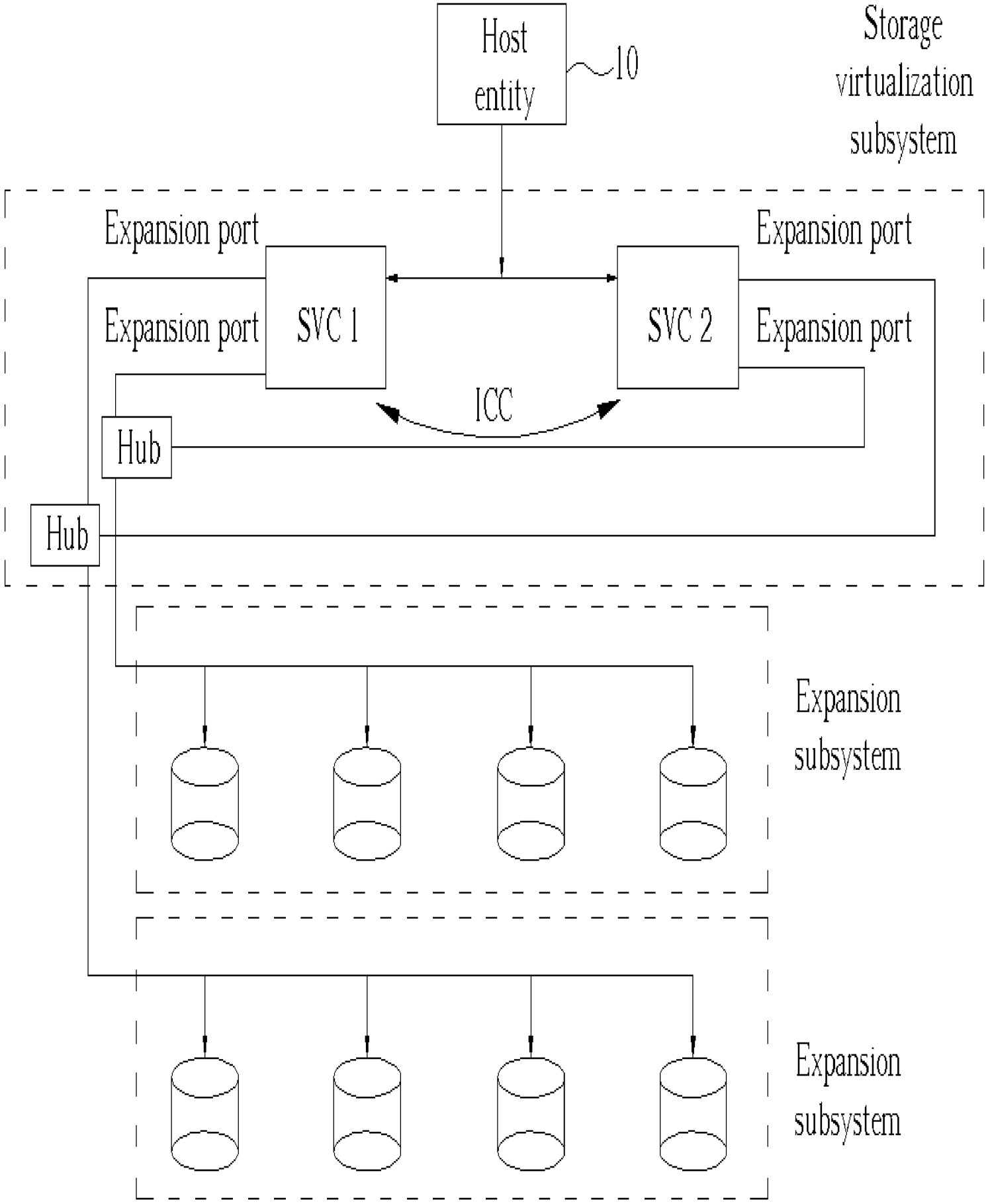

Host entity
10
Storage virtualization subsystem
Expansion port
Expansion port
Expansion port
SVC 1
SVC 2
Expansion port
ICC
Hub
Hub
Expansion subsystem
Expansion subsystem

Fig. 21

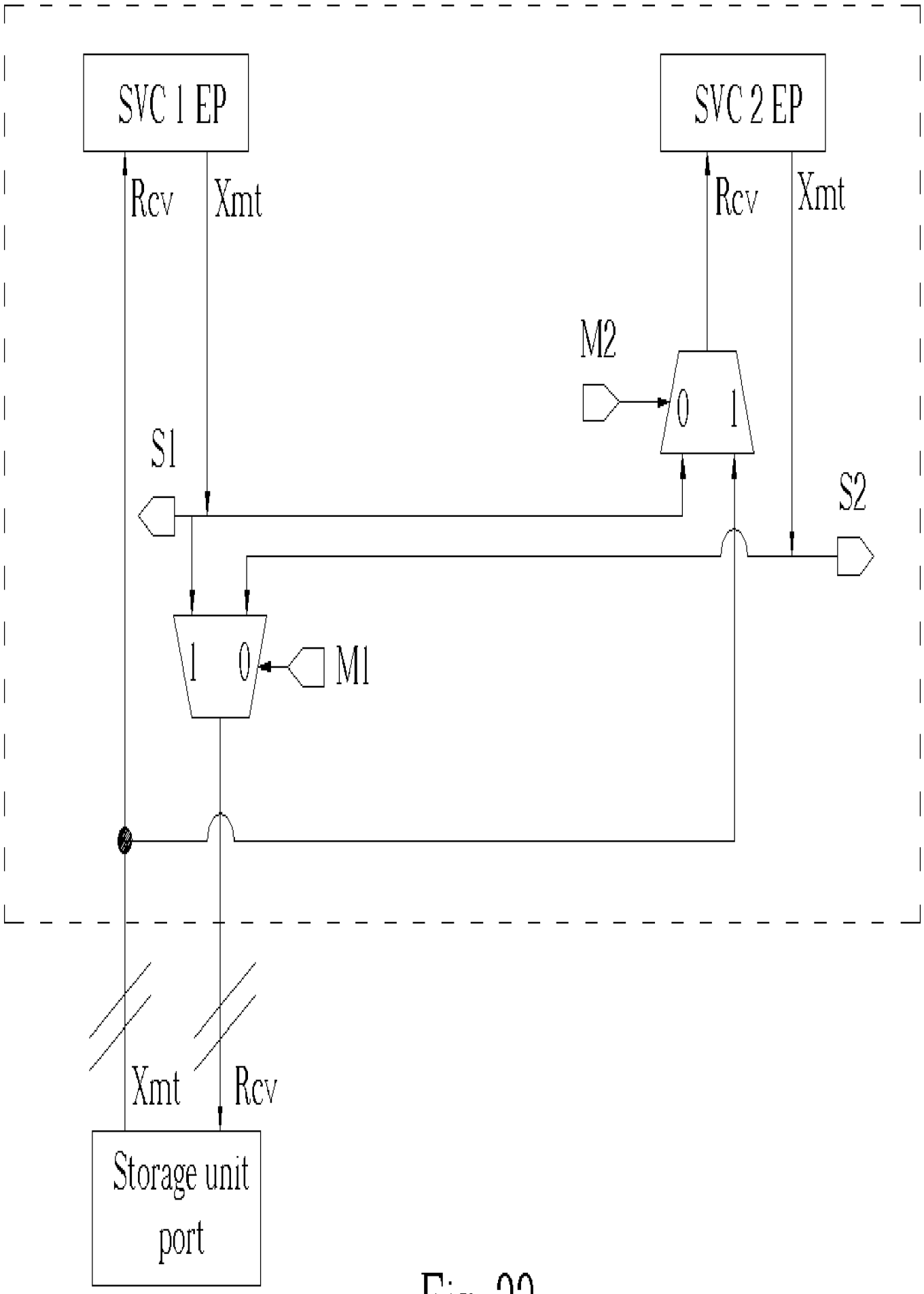

SVC 1 EP
SVC 2 EP
Rcv
Xmt
Rcv
Xmt
M2
0
1
S1
S2
1
0
M1
Xmt
Rcv
Storage unit port

Fig. 22

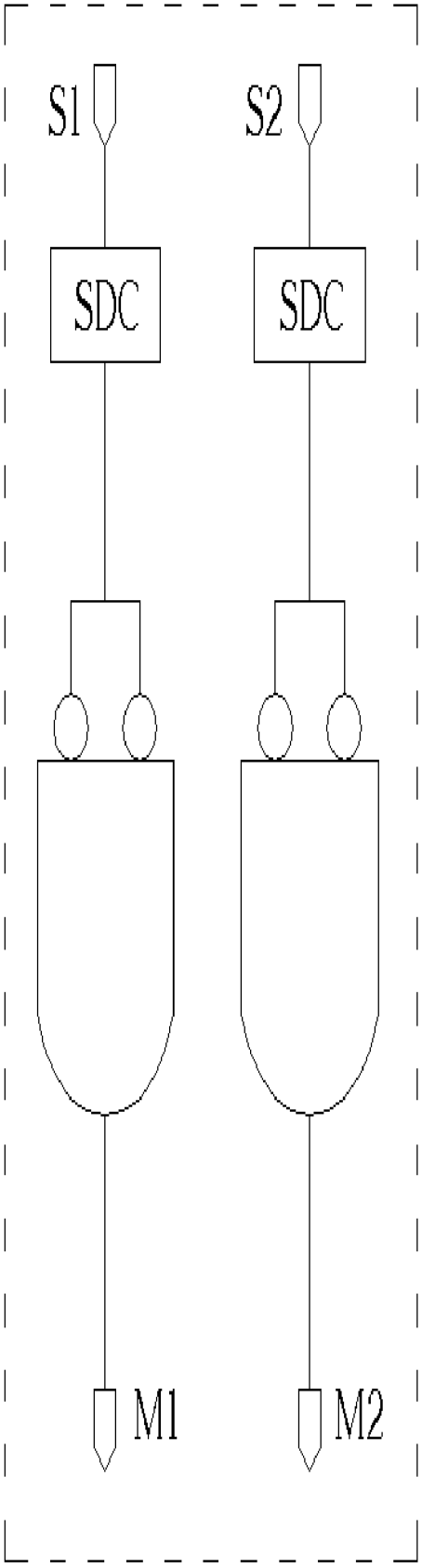


Fig. 23

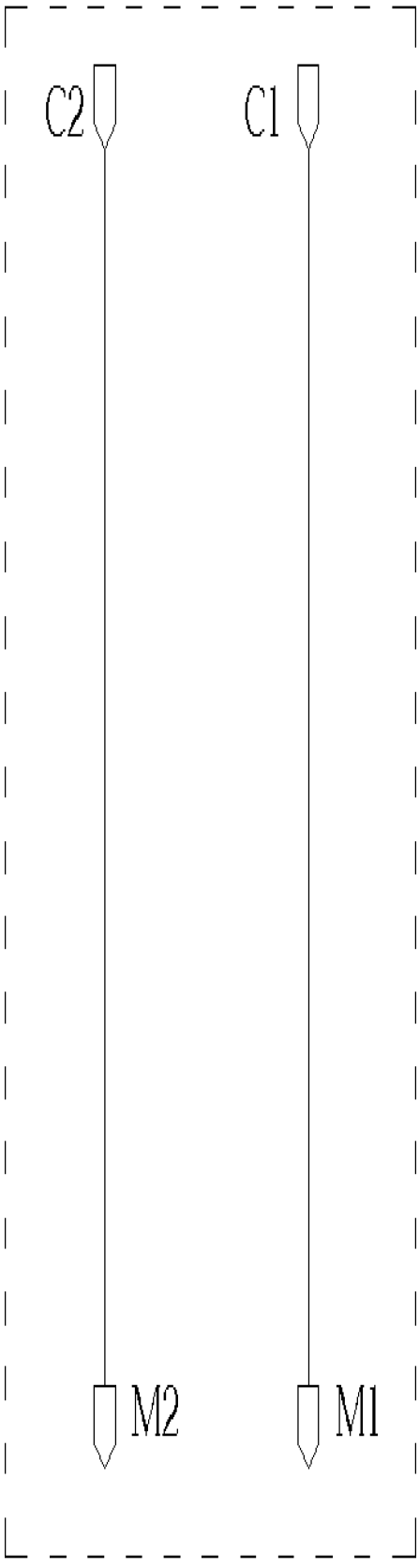


Fig. 24

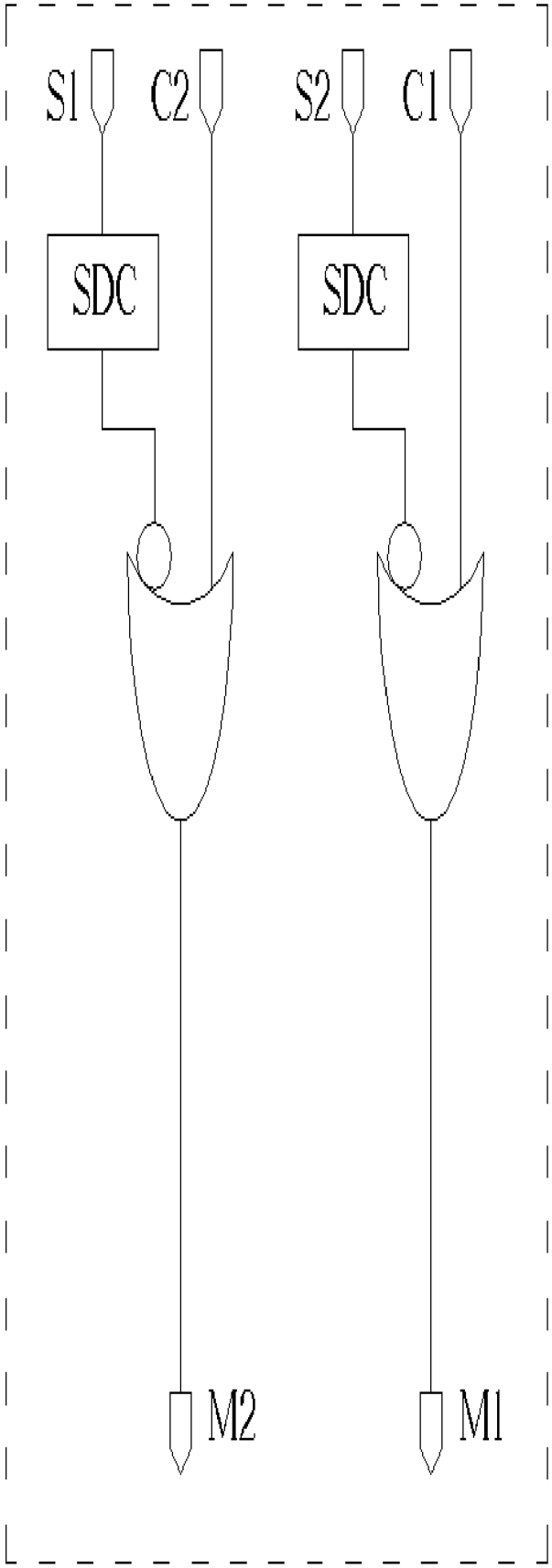


Fig. 25

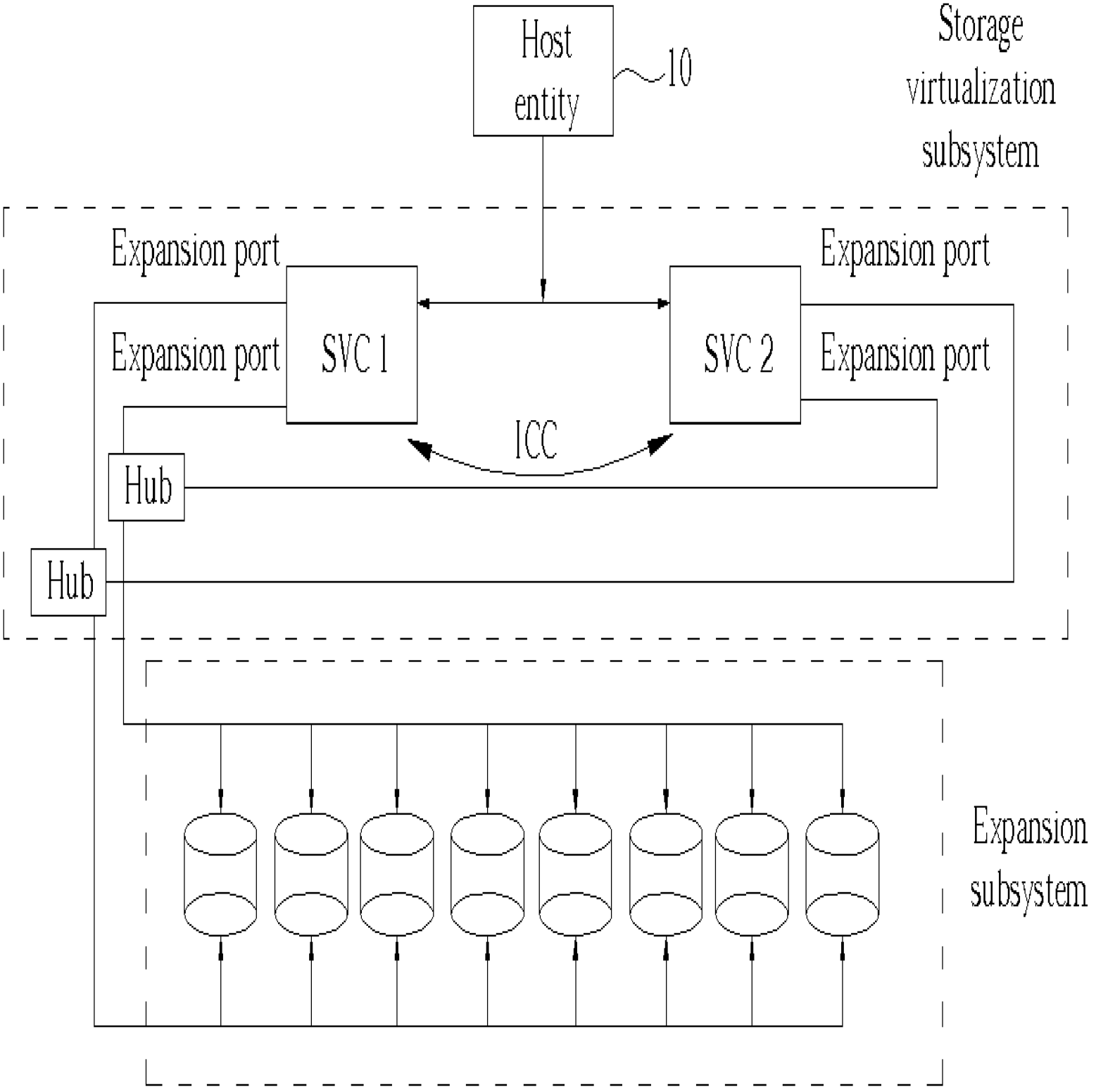

Host entity
10
Storage virtualization subsystem
Expansion port
Expansion port
Expansion port
Expansion port
SVC 1
SVC 2
ICC
Hub
Hub
Expansion subsystem

Fig. 26

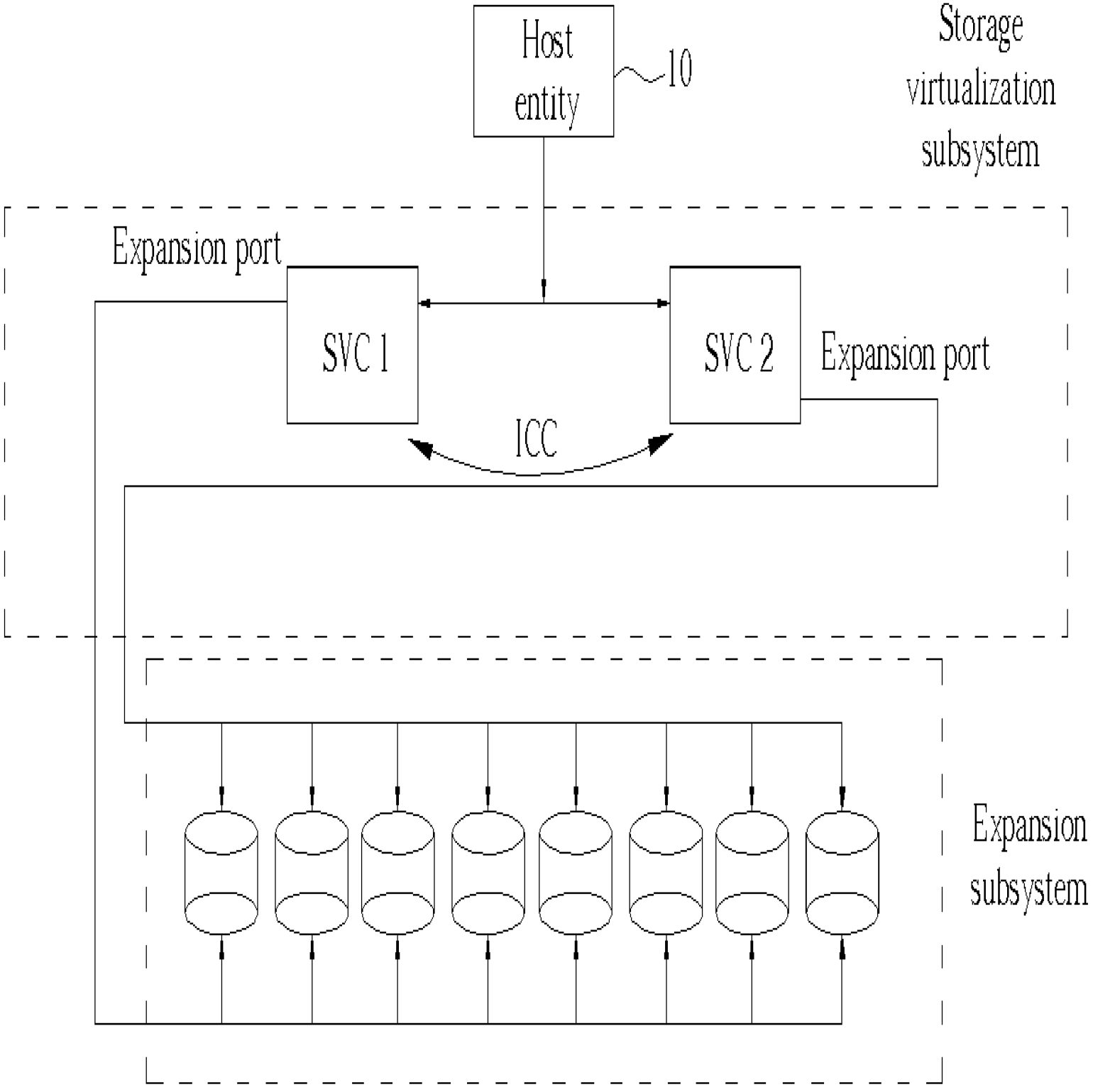

Host entity
10
Storage virtualization subsystem
Expansion port
SVC 1
SVC 2
Expansion port
ICC
Expansion subsystem

Fig. 27

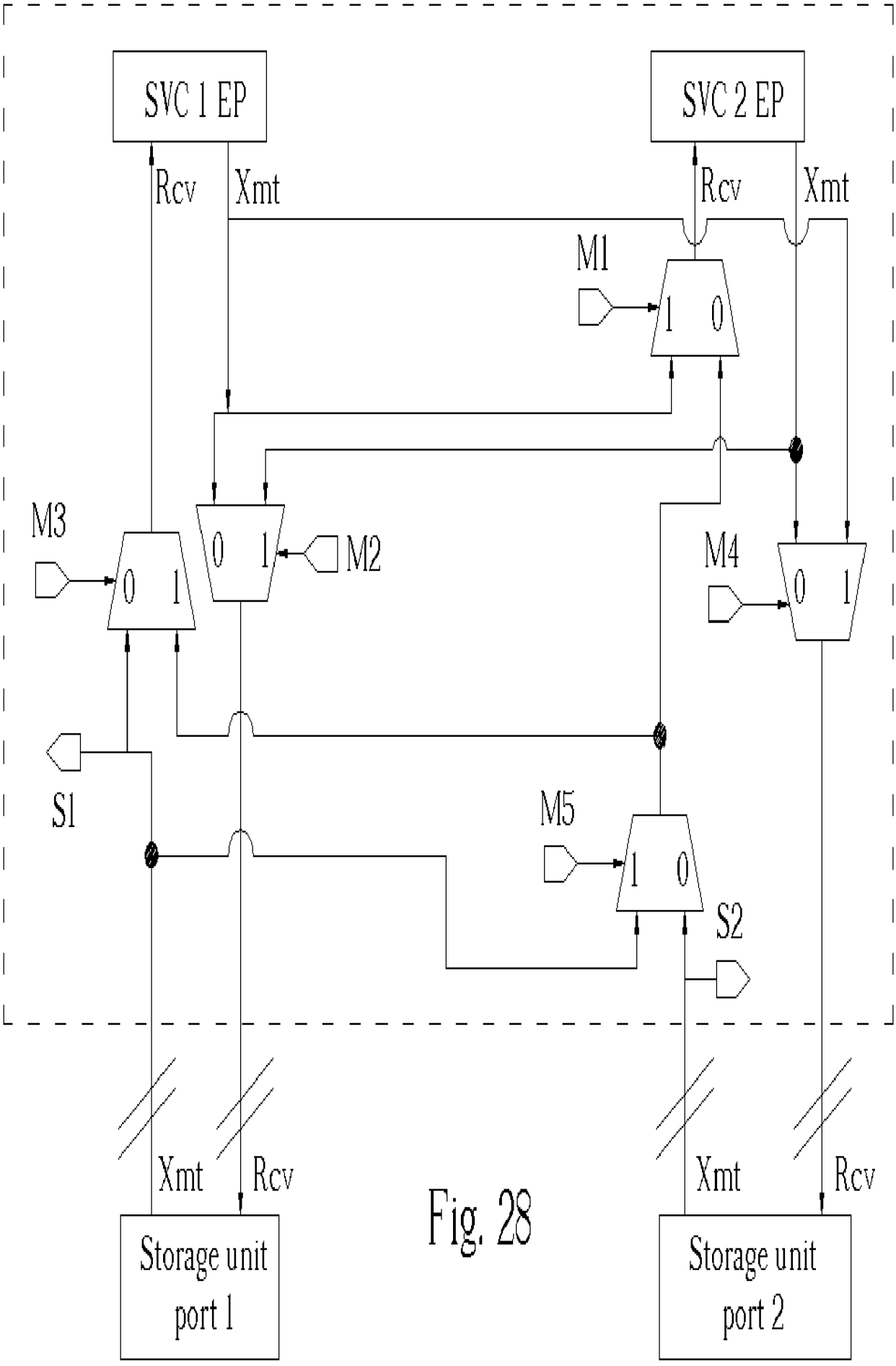

SVC 1 EP
SVC 2 EP
Rcv
Xmt
Rcv
Xmt
M1
1 0
M3
0 1
0 1
M2
M4
0 1
S1
M5
1 0
S2
Xmt
Rcv
Xmt
Rcv
Storage unit port 1
Storage unit port 2

Fig. 28

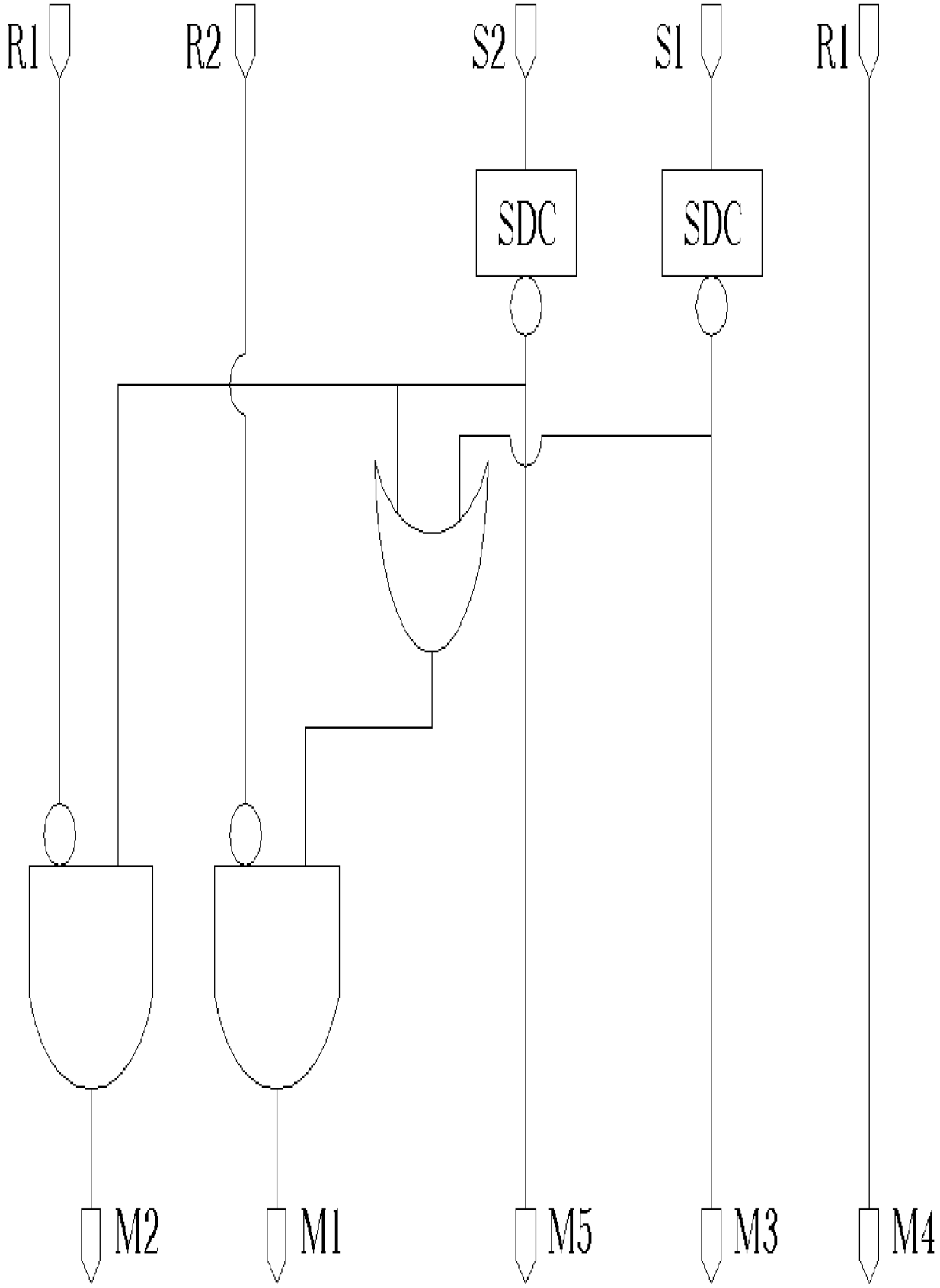


Fig. 29

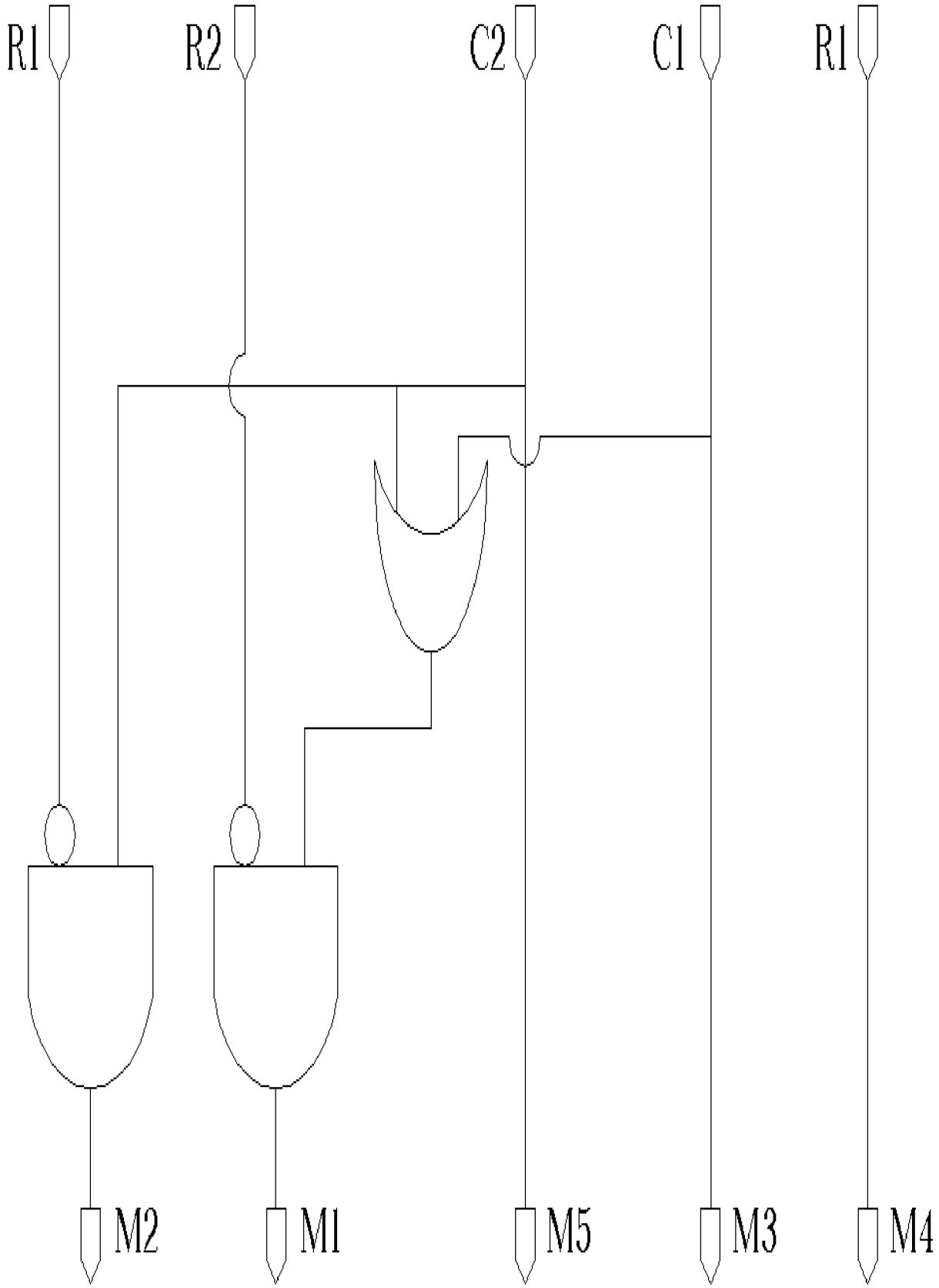


Fig. 30

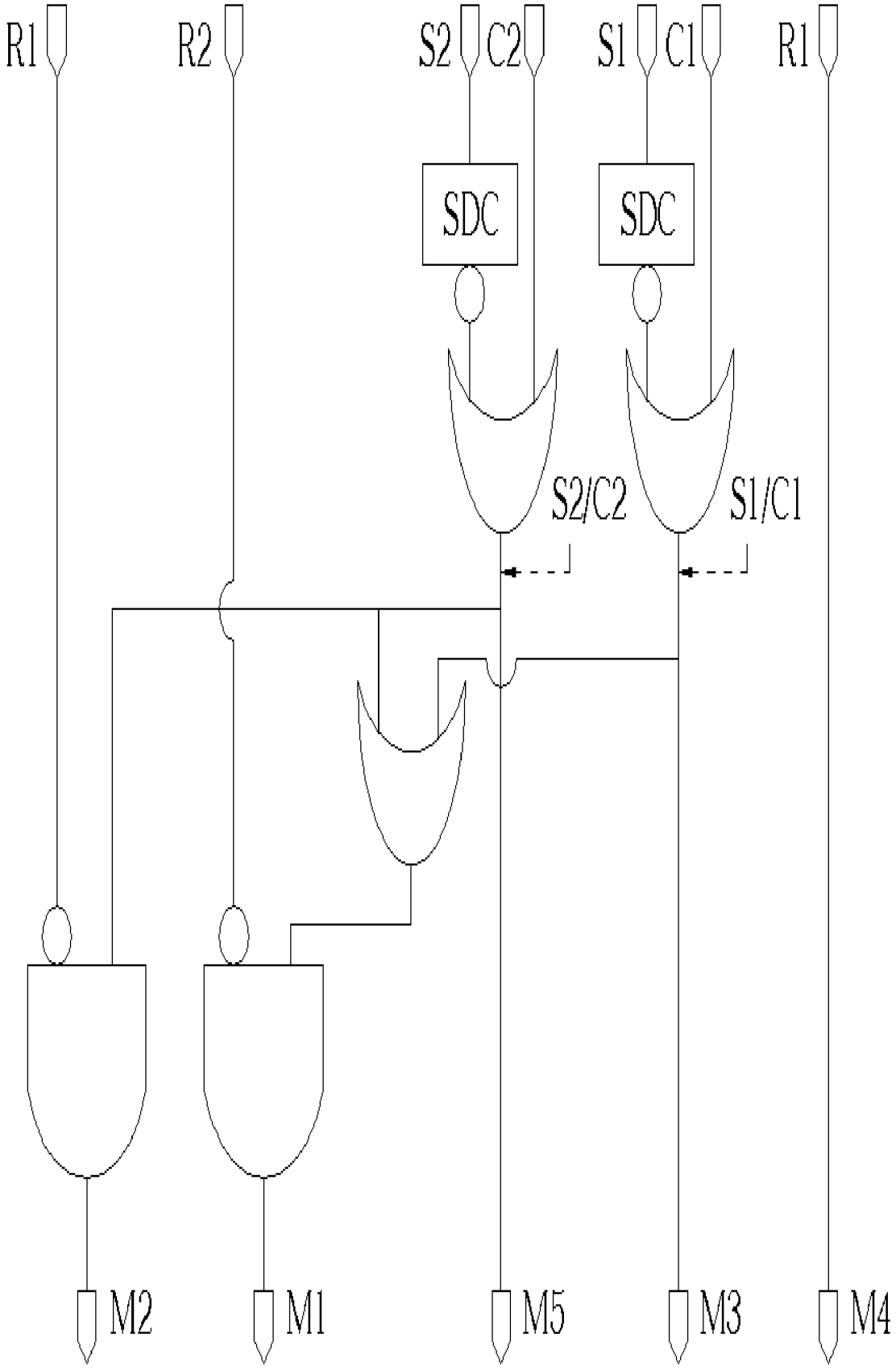


Fig. 31

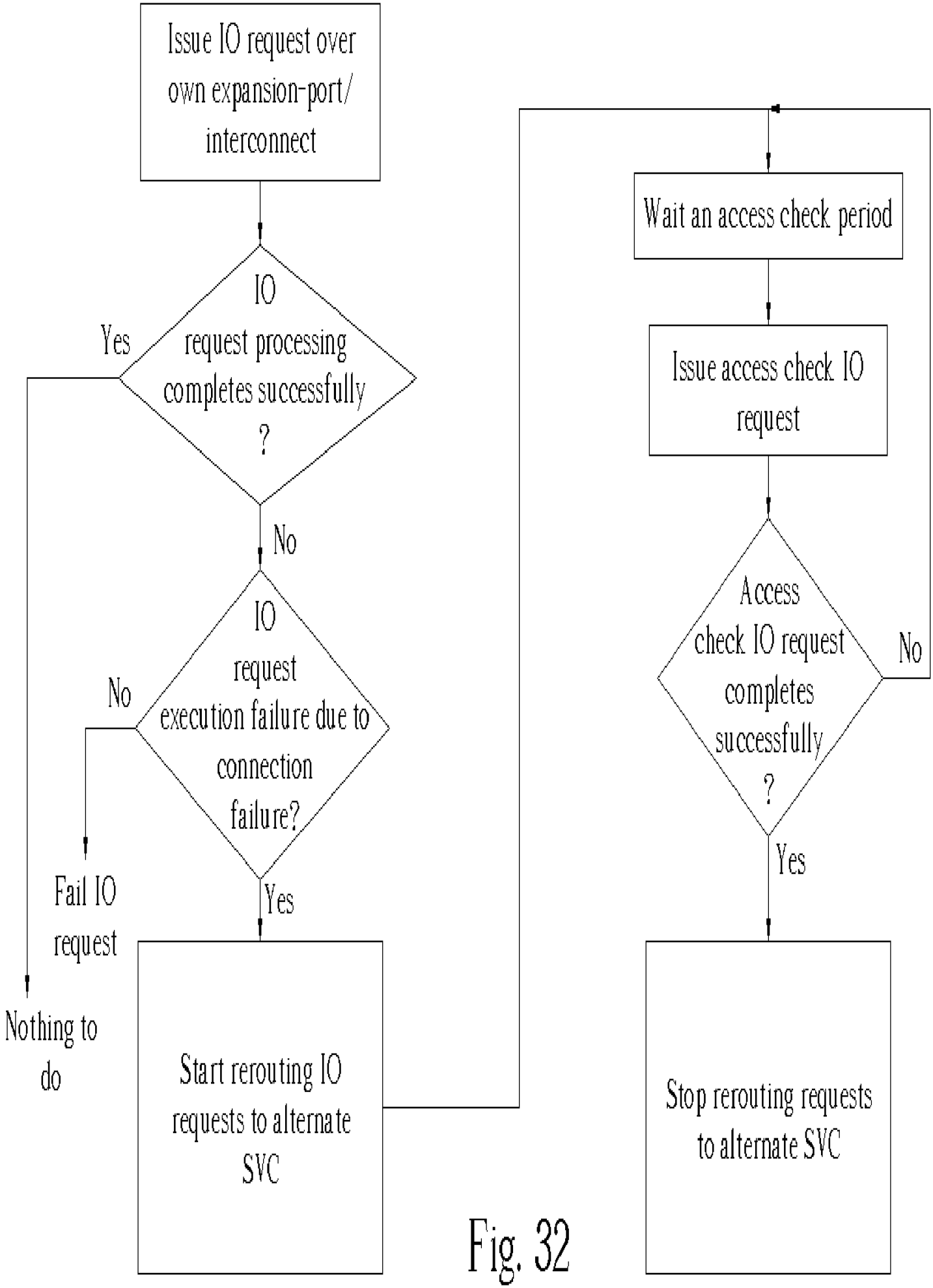

Issue IO request over own expansion-port/ interconnect
IO request processing completes successfully ?
Yes
No
IO request execution failure due to connection failure?
No
Yes
Fail IO request
Nothing to do
Start rerouting IO requests to alternate SVC
Wait an access check period
Issue access check IO request
Access check IO request completes successfully ?
No
Yes
Stop rerouting requests to alternate SVC

Fig. 32

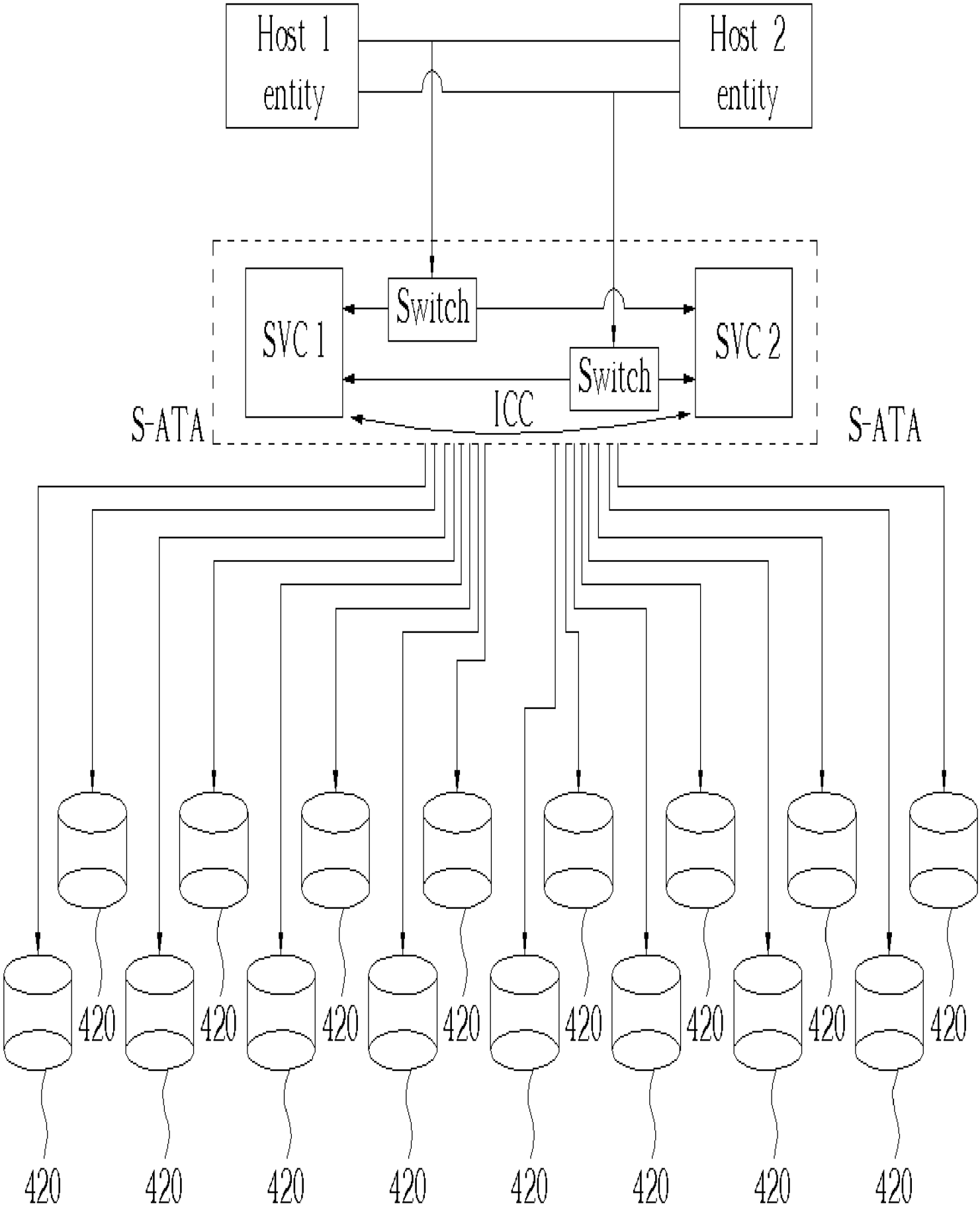

Host 1 entity
Host 2 entity
SVC 1
Switch
Switch
SVC 2
ICC
S-ATA
S-ATA
420

Fig. 33

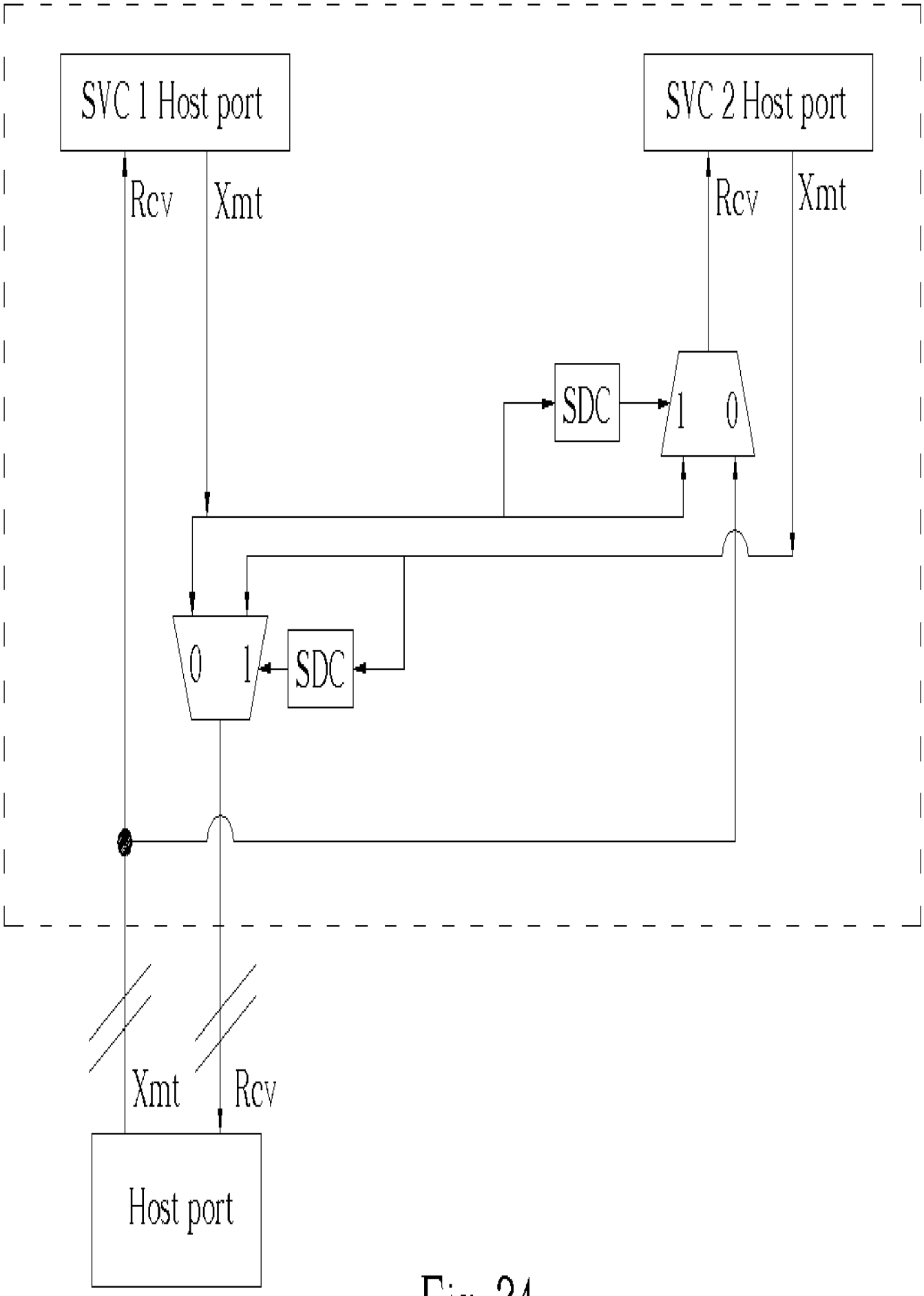

SVC 1 Host port
SVC 2 Host port
Rcv
Xmt
Rcv
Xmt
SDC
1
0
0
1
SDC
Xmt
Rcv
Host port

Fig. 34

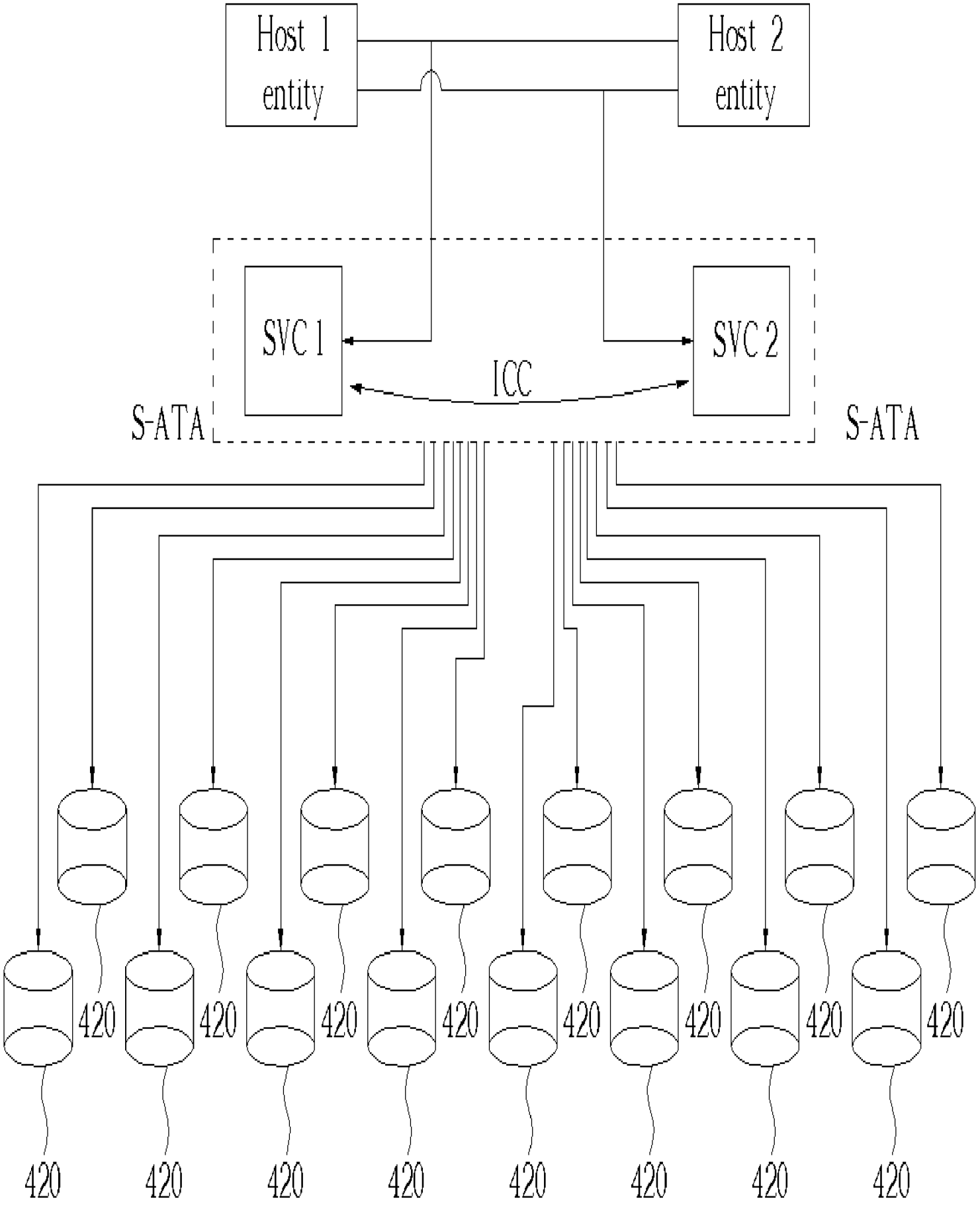

Host 1 entity
Host 2 entity
SVC 1
SVC 2
ICC
S-ATA
S-ATA
420
420
420
420
420
420
420
420
420
420
420
420
420
420
420
420

Fig. 35

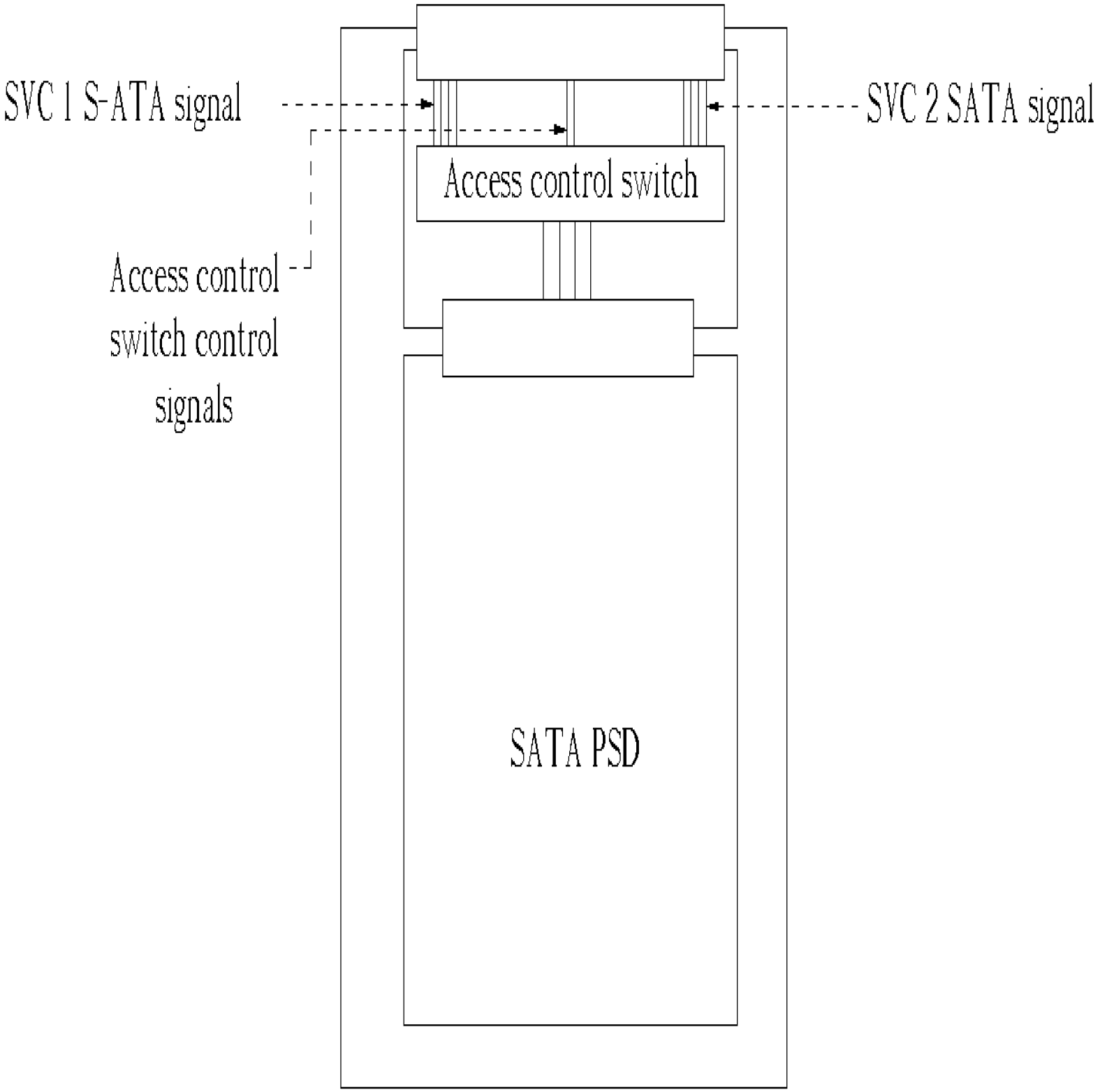

SVC 1 S-ATA signal
SVC 2 SATA signal
Access control switch
Access control switch control signals
SATA PSD

Fig. 36

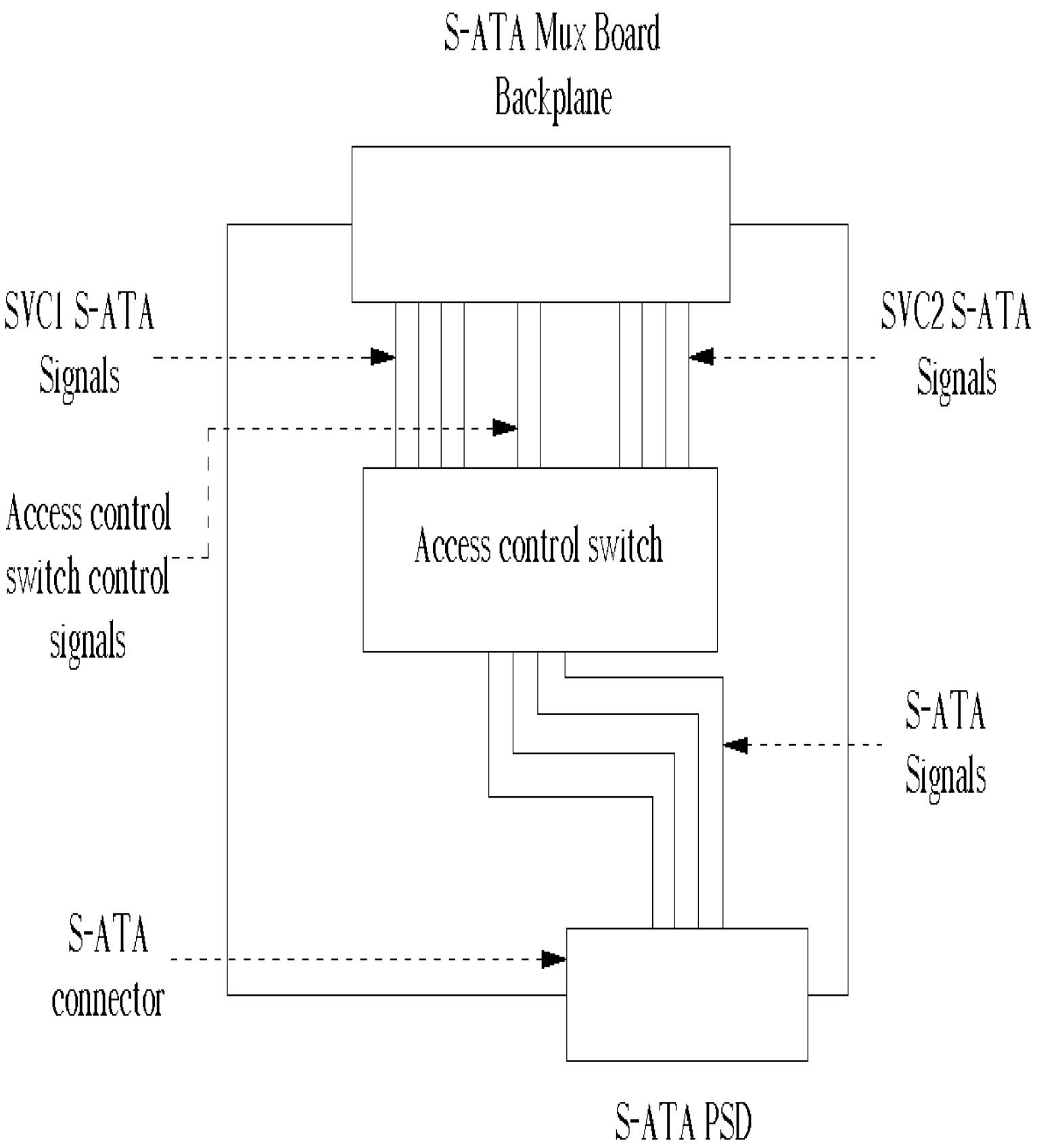

S-ATA Mux Board
Backplane
SVC1 S-ATA
Signals
SVC2 S-ATA
Signals
Access control
switch control
signals
Access control switch
S-ATA
Signals
S-ATA
connector
S-ATA PSD

Fig. 37

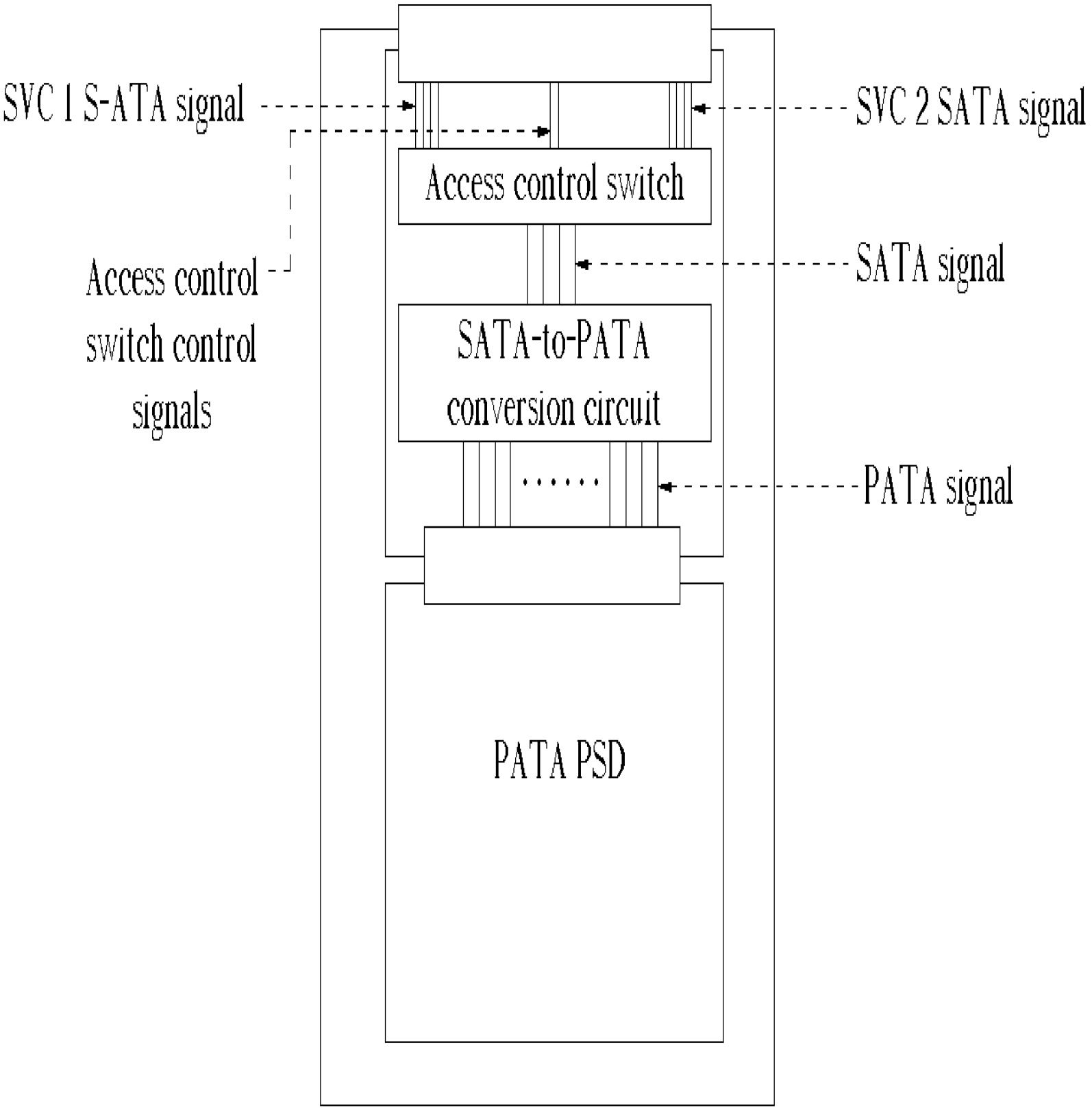

SVC 1 S-ATA signal
SVC 2 SATA signal
Access control switch
Access control switch control signals
SATA signal
SATA-to-PATA conversion circuit
.....
PATA signal
PATA PSD

Fig. 38

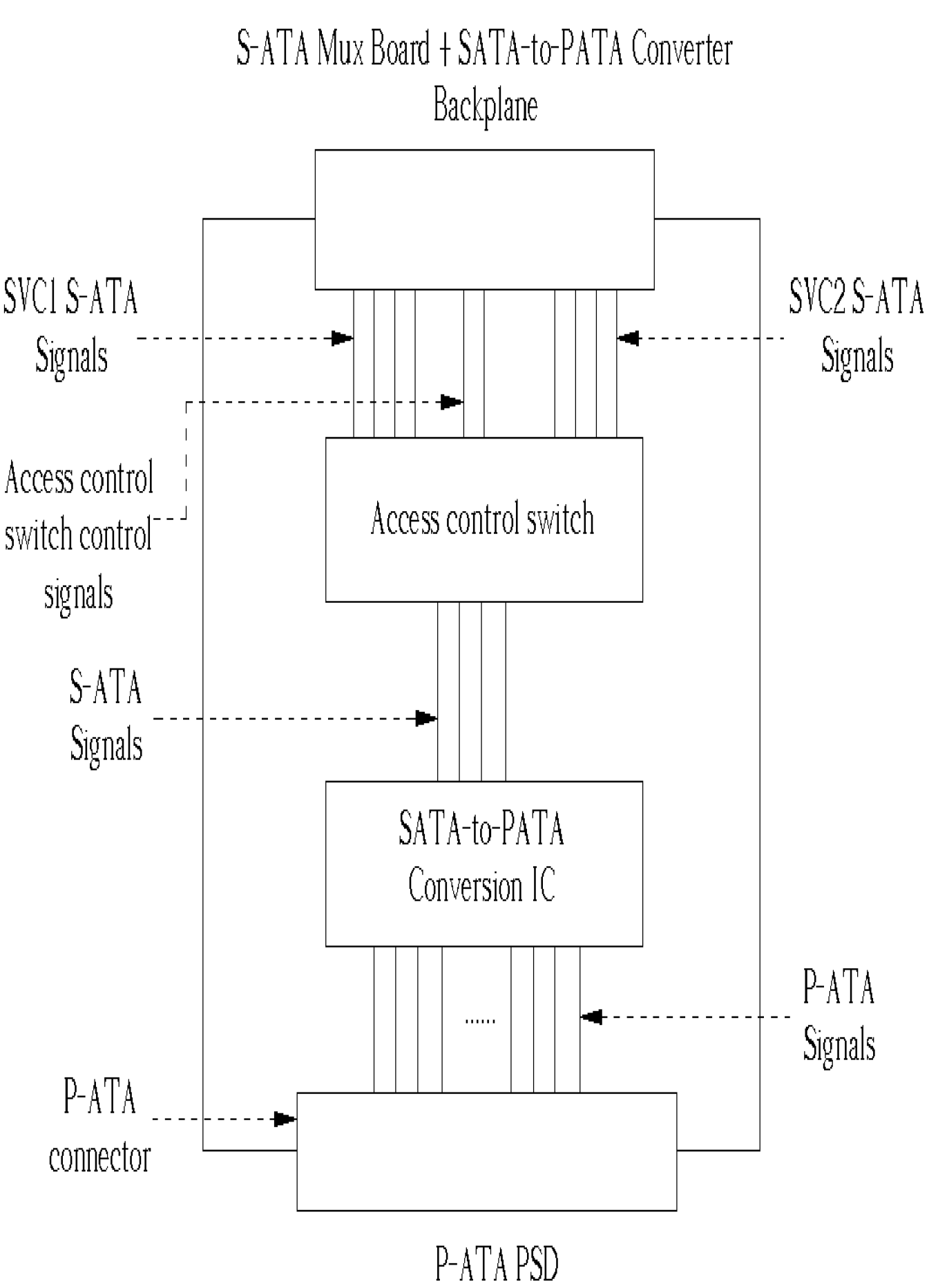

S-ATA Mux Board + SATA-to-PATA Converter
Backplane
SVC1 S-ATA Signals
SVC2 S-ATA Signals
Access control switch control signals
Access control switch
S-ATA Signals
SATA-to-PATA Conversion IC
.....
P-ATA Signals
P-ATA connector
P-ATA PSD

Fig. 39

CS1	CS2	CS	SI1->PI	SI2->PI
0	0	0	Closed	Open
0	Tri	0	Closed	Open
Tri	0	0	Closed	Open
Tri	Tri	1	Open	Closed

Fig. 40

CS1	CS2	SI1->PI	SI2->PI	Note
0	0	Undef	Undef	*:most recent changed one of CS1, CS2 gets access.
0	1	Open	Closed	
1	0	Closed	Open	
1	1	*Note	*Note	

Fig. 41

Loop Connections	M1	M2	M3	M4	M5
C1<=>P1,C2<=>P2	0	0	0	0	0
C1=>C2=>P1=>C1	1	1	0	DC	DC
C1=>C2=>P2=>C1	1	DC	1	0	0
C1<=>P1	DC	0	0	DC	DC
C1<=>P2	DC	DC	1	1	0
C2<=>P2	0	DC	DC	0	0
C2<=>P1	0	1	DC	DC	1

Cn: SVCn, Pn: Storage Unit Port n; n=1, 2.

Fig. 42

S1	S2	R1	R2	M1	M2	M3	M4	M5
Val	Val	0	0	0	0	0	0	0
Inv	Val	0	0	1	0	1	0	0
Val	Inv	0	0	1	1	0	0	1
Val	Val	1	0	0	0	0	1	0
Inv	Val	1	0	1	0	1	1	0
Val	Inv	1	0	1	0	0	1	1
Val	Val	0	1	0	0	0	0	0
Inv	Val	0	1	0	0	1	0	0
Val	Inv	0	1	0	1	0	0	1

Fig. 43

C1	C2	R1	R2	M1	M2	M3	M4	M5
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	1	0	0
0	1	0	0	1	1	0	0	1
0	0	1	0	0	0	0	1	0
1	0	1	0	1	0	1	1	0
0	1	1	0	1	0	0	1	1
0	0	0	1	0	0	0	0	0
1	0	0	1	0	0	1	0	0
0	1	0	1	0	1	0	0	1

Fig. 44

S1/C	S2/C	R1	R2	M1	M2	M3	M4	M5
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	1	0	0
0	1	0	0	1	1	0	0	1
0	0	1	0	0	0	0	1	0
1	0	1	0	1	0	1	1	0
0	1	1	0	1	0	0	1	1
0	0	0	1	0	0	0	0	0
1	0	0	1	0	0	1	0	0
0	1	0	1	0	1	0	0	1

Fig. 45